

8-2014

A Study of Automatic Detection and Classification of EEG Epileptiform Transients

Jing Zhou

Clemson University, zhou5@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Zhou, Jing, "A Study of Automatic Detection and Classification of EEG Epileptiform Transients" (2014). *All Dissertations*. 1275.
https://tigerprints.clemson.edu/all_dissertations/1275

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

A STUDY OF AUTOMATIC DETECTION AND CLASSIFICATION OF EEG EPILEPTIFORM TRANSIENTS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

by
Jing Zhou
August 2014

Accepted by:
Dr. Robert Schalkoff, Committee Chair
Dr. John Gowdy
Dr. Brian Dean
Dr. Carl Baum
Dr. Jonathan Halford

Abstract

This dissertation documents methods for automatic detection and classification of epileptiform transients, which are important clinical issues. There are two main topics: (1) Detection of paroxysmal activities in EEG; and (2) Classification of paroxysmal activities. This machine learning algorithms were trained on expert opinion which was provided as annotations in clinical EEG recordings, which are called “yellow boxes” (YBs).

The dissertation describes improved wavelet-based features which are used in machine learning algorithms to detect events in clinical EEG. It also reveals the influence of electrode positions and cardinality of datasets on the outcome. Furthermore, it studies the utility of using fuzzy strategies to obtain better performance than using crisp decision strategies.

In the yellow-box detection study, this dissertation makes use of threshold strategies and implementation of ANNs. It develops two types of features, wavelet and morphology, for comparison. It also explores the possibility to reduce input vector dimension by pruning. A full-scale real-time simulation of YB detection is performed. The simulation results are demonstrated using a web-based EEG viewing system designed in the School of Computing at Clemson, called EEGnet. Results are compared to expert marked YBs.

Acknowledgments

Many people contributed time, knowledge, skill, and support to this research. I am pleased to acknowledge their contributions. First, I would like to thank Dr. Jonathan Halford for supporting us with all the materials needed in this project. And I would like to thank Dr. Schalkoff, Dr. Gowdy, Dr. Dean and Dr. Baum for serving as committee members. I would also like to thank Dr. Dean and his students for preparing data for us.

Table of Contents

Title Page	i
Abstract	ii
Acknowledgments	iii
List of Tables	vii
List of Figures	ix
1 Introduction	1
1.1 Problem Background	1
1.2 Previous Related Work	2
1.3 Purpose of the Research and Overview of the Dissertation	4
2 Data Acquisition and Research Methods	6
2.1 Routine Scalp EEG and the International 10-20 System	6
2.2 Data Acquisition	10
2.2.1 Dataset Based on Crisp Scoring	10
2.2.2 Dataset Based on Fuzzy Scoring	14
2.3 Multi-Resolution Analysis and Discrete Wavelet Transform	15
2.4 Energy Distribution of EEG Signal by Wavelet Transform	21
3 Classification of Expert-Marked Yellow-Boxes	23
3.1 Crisp Classification	23
3.1.1 Methodology for Design of Classification of Expert-Marked Yellow-Boxes	23
3.1.1.1 Benchmark Wavelet Feature Set	24
3.1.1.2 Feature Selection	25
3.1.1.3 Employment of Multiple Mother Wavelets	26
3.1.1.4 Scalp Spatial Features	26
3.1.1.5 Methodology of Classification and Performance Evaluation	27
3.1.1.5.1 Balance of the Dataset	28
3.1.1.5.2 k-Nearest Neighbor Rule	29
3.1.1.5.3 k-Fold Cross-Validation	29
3.1.1.5.4 Performance Evaluation	30
3.1.2 Results and Evaluation of Yellow-Box Classification	31
3.1.2.1 Comparison of Performances on Selected Feature Set	31
3.1.2.2 Max vs All	31
3.1.2.3 Effects of Electrode Pair Scalp Location Features	36
3.1.2.4 Effects of the Size of the Dataset	36
3.1.2.5 Statistic Significance of Detection Improvement	37

	3.1.2.5.1	One-Tailed t-Test	37
	3.1.2.5.2	Power of the Test	39
3.2	Fuzzy Classification		41
	3.2.1	Fuzzy k-Nearest-Neighbor Algorithm	41
	3.2.2	Fuzzy c-Means	42
	3.2.3	Initialization of the Membership Function	43
	3.2.3.1	Means	44
	3.2.3.1.1	Arithmetic Mean	44
	3.2.3.1.2	Geometric Mean	44
	3.2.3.1.3	Cube Root of the Cubes' Mean	44
	3.2.3.1.4	N th Root of the N th Powers' Mean	45
	3.2.3.2	Histogram Equalization	45
	3.2.3.3	Interpolation and Polynomial Fitting	48
	3.2.3.3.1	Interpolation	48
	3.2.3.3.2	Polynomial Fitting	50
	3.2.3.4	Function Based Initialization	50
	3.2.3.4.1	Linear Normalization of the "Votes" on the paroxysmal Type	50
	3.2.3.4.2	Sigmoid Initialization	51
	3.2.3.4.3	Synthetic Function Based on Confidence Factor and "vote"	51
	3.2.3.4.4	Multi-Dimension Function Application	51
	3.2.3.5	Biased Confidence Factor	52
	3.2.3.6	Optimization of the Coefficients Using Gradient Descent	54
3.2.4	Performance on Fuzzy Set		56
	3.2.4.1	Fuzzy k-Nearest-Neighbor	56
	3.2.4.1.1	Benchmark of Crisp k-NNR	56
	3.2.4.1.2	Results of the Fuzzy k-NNR	56
	3.2.4.2	Fuzzy c-Means	69
	3.2.4.2.1	Clustering Results of Fuzzy c-means	69
	3.2.4.2.2	Comparison to Crisp c-Means	71
4	Yellow-Box Detection		72
4.1	Methodology for Design of Detection of Yellow-Box		72
	4.1.1	Plain Detection	73
	4.1.1.1	Synopsis of Indiradevi's Algorithm (2007)	74
	4.1.1.2	Subband Weight Optimization by Belief Value	76
	4.1.1.3	Methodology of Performance Evaluation of Plain Detection	76
	4.1.2	Implementation of Artificial Neural Network with a Pruning Procedure	77
	4.1.2.1	Calculation of S_{ij} , the Sensitivity of the Error Function	80
	4.1.2.2	Dingle's Feature Choice: Morphology and Background	82
	4.1.3	Clustering of Yellow-Boxes	84
	4.1.3.1	Grouping	84
	4.1.3.2	Merging and Discarding	86
	4.1.4	Full-scale Real-time Simulation	86
4.2	Results and Evaluation of Yellow-Box Detection		88
	4.2.1	Implementation of Indiradevi's Algorithm	88
	4.2.2	Implementation of ANN	95
	4.2.3	Interpretation of S_{ij} and Implementation of ANN with Pruning Strategy	96
	4.2.3.1	S_{ij} of Input Layer's Weights	96
	4.2.3.2	Confirmation of Results with Feature/Subband Pruning	100
	4.2.3.3	Re-Test of ANN after Pruning	101
	4.2.4	Implementation of ANN with Morphological Features	102

4.2.5	Performance of Full-scale Real-time Simulation	103
5	Conclusions and Discussion	117
5.1	Yellow-Box Classification	117
5.1.1	Performance of crisp classification	117
5.1.2	Performance of fuzzy classification	118
5.2	Yellow-Box Detection	127
5.3	Future Research	129
Appendices	130
A	Matlab Code Structure	131

List of Tables

2.1	Distribution of the ‘phase2’ confidence factor values by “votes” in paroxysmal types	14
2.2	Distribution of the ‘phase2a’ confidence factor values by “votes” in paroxysmal types	14
3.1	Corresponding frequency range of each subband in classification	24
3.2	Feature choices and dimensions of new feature sets	26
3.3	Coordinate information of electrode channels	28
3.4	k-NNR (k=3) comparative classification results of new feature sets	34
3.5	k-NNR (k=3) classification results of overall features vs. maxima	34
3.6	k-NNR (k=3) classification results with/without location features	35
3.7	k-NNR (k=3) classification results with datasets of different size	35
3.8	The highest level at which H_0 can be rejected with different feature/wavelet choices	38
3.9	The highest level at which H_0 can be rejected of single vs. double mother wavelets .	38
3.10	Power with a level of significance of 0.05 (different wavelet choices)	40
3.11	Power with a level of significance of 0.05 (single vs. double mother wavelets)	40
3.12	Customization of the coefficient of a confidence factor based on votes	53
3.13	Crisp classification result on 200-patient dataset and selected subset	57
3.14	Confidence factors before and after equalization and interpolation in Condition6 . . .	59
3.15	The coefficients of the 3rd order polynomial in Condition7	60
3.16	The coefficients of vote-based confidence factor in Condition14	61
3.17	The coefficients of vote-based confidence factor in Condition15	61
3.18	The coefficients of vote-based confidence factor in Condition16	61
3.19	The coefficients of gradient descent optimization in Condition17	62
3.20	Results of the fuzzy k-NNR based tests on the 200-patient dataset	63
3.21	Comparison between selected crisp and fuzzy results	69
3.22	Number of data vectors related with <i>cth</i> mean in 2-mean case on ‘phase2’	70
3.23	Number of data vectors related with <i>cth</i> mean in 3-mean case on ‘phase2’	70
3.24	Number of data vectors related with <i>cth</i> mean in 2-mean case on 200-p set	70
3.25	Number of data vectors related with <i>cth</i> mean in crisp 2-mean case on ‘phase2’ . . .	70
3.26	Number of data vectors related with <i>cth</i> mean in crisp 3-mean case on ‘phase2’ . . .	71
3.27	Number of data vectors related with <i>cth</i> mean in 2-mean case on 200-p set	71
4.1	Corresponding frequency range of each subband in detection	75
4.2	Detector results using equal weight	91
4.3	Detector results of AND decision by D_4 & D_5	92
4.4	Detector results of optimal weight #1	93
4.5	Detector results of optimal weight #2	94
4.6	Summary of Indiradevi’s algorithm	95
4.7	ANN performances in YB detection	97
4.8	Performance of ANN with restricted input	100
4.9	Comparison of the ANN performances with features from subband D1 eliminated . .	101
4.10	ANN performance in YB detections after pruning of input features	102

4.11 ANN performance in YB detections using morphological features	102
4.12 Simulation results without grouping	104
4.13 Simulation results grouped with overlap rates of 50%	104
4.14 Simulation results grouped with overlap rates of 1e-3%	105
4.15 Morphology feature based simulation results without grouping	105
4.16 Morphology feature based simulation results grouped with overlap rates of group50%	105
A.1 Selected Matlab Code	134

List of Figures

1.1	The flow chart of the desired ETs detection system	4
1.2	The flow chart of the detection module	5
1.3	The flow chart of the classification module	5
2.1	21-electrode International 10-20 system	7
2.2	Lobes in hemisphere	7
2.3	Electrode distance	8
2.4	Interface of EEGNet	11
2.5	10-second rsEEG data segment in EEGNet	11
2.6	Abnormal Epileptiform PED	12
2.7	Artifact PED	12
2.8	Normal Electroconvulsive PED	12
2.9	Wavelet decomposition tree	20
3.1	Comparison between wavelet functions and ETs	24
3.2	Sample EEG wavelet decomposition results using DB4 and DB2	27
3.3	Composite summary of feature set evaluations	32
3.4	k-NNR (k=3) comparative classification results of new feature sets	33
3.5	Histogram of the 200-patient dataset before and after equalization	47
3.6	Relation between the renewed confidence factor values and their original counterpart	48
3.7	Spline/pchip interpolation	49
3.8	Strategies to determine ground truth for the 200-patient dataset	58
3.9	The trend of the biased coefficients in Condition16	62
4.1	Module of the spike detector	72
4.2	Multilayer feedforward network using back-propagation algorithm	78
4.3	ANN performance with different ratio of training data	98
4.4	The proportion of values of S_{ij} for selected input	99
4.5	Yellow boxes annotated by expert on Patient#1	107
4.6	Raw yellow box candidates annotated by Simulation#2 on Patient#1	108
4.7	Yellow box annotated by Simulation#2 on Patient#1 after grouping	109
4.8	Raw yellow box candidates annotated by Simulation#9 on Patient#1	110
4.9	Yellow box annotated by Simulation#9 on Patient#1 after grouping	111
4.10	Yellow boxes annotated by expert on Patient#5	112
4.11	Raw yellow box candidates annotated by Simulation#2 on Patient#5	113
4.12	Yellow box annotated by Simulation#2 on Patient#5 after grouping	114
4.13	Raw yellow box candidates annotated by Simulation#9 on Patient#5	115
4.14	Yellow box annotated by Simulation#9 on Patient#5 after grouping	116
5.1	Exemplar of biased weights (Condition16) with k = 1	120
5.2	Exemplar of biased weights (Condition16) with k = 3	121
5.3	Exemplar of biased weights (Condition16) with k = 5	121

5.4	Exemplar of biased weights (Condition16) with $k = 7$	122
5.5	Exemplar of biased weights (Condition16) with $k = 9$	122
5.6	Energy of error of biased weights (Condition16) with different choice of 'k'	123
5.7	Energy of error of Condition16	123
5.8	Exemplar of sigmoid transfer (Condition8) with $k = 1$	124
5.9	Exemplar of sigmoid transfer (Condition8) with $k = 3$	124
5.10	Exemplar of sigmoid transfer (Condition8) with $k = 5$	125
5.11	Exemplar of sigmoid transfer (Condition8) with $k = 7$	125
5.12	Exemplar of sigmoid transfer (Condition8) with $k = 9$	126
5.13	Energy of error of sigmoid transfer (Condition8) with different choice of 'k'	126
5.14	Energy of error of Condition8	127
A.1	Code Structure of Classification	131
A.2	Code Structure of Detection	132

Chapter 1

Introduction

1.1 Problem Background

Epilepsy is characterized by sudden recurrent and transient disturbances of mental function or movements of the body that result from paroxysmal and abnormal discharge of groups of brain cells [37] [45]. It is the second most common neurological disorder (after stroke) [65]. Approximately one percent of the world population has epilepsy [28]. The most common clinical procedure in epilepsy related diagnosis is the routine scalp electroencephalogram (rsEEG) recording, which is a summation of electrical activities generated by cortical neurons along the scalp [66] [27]. Epileptiform transients (ETs) are brief bursts of activity (usually lasting less than one second which occur intermittently throughout the day and night in patients with epilepsy). ETs appear in the EEG in the form of spikes (last 20-70 ms) or sharp waves (last 70-200 ms) with pointed peaks. Some ETs have a more complex form: a spike followed by a slow wave (lasts 150-350 ms), together called spike-and-slow-wave-complex [34]. For a patient who is having seizure-like events, the presence of ETs is a sign the patient may have one or more seizures in the future [60]. Therefore detection of ETs is very useful in the diagnosis of epilepsy [20].

ETs are usually detected by visual inspection by experienced physicians. This process is notoriously time consuming, especially in the case of long term EEG recordings, e.g. 24-hour continuous ambulatory monitoring studies. In addition, there is considerable variability in the detection of ETs in EEG by physicians [40] [30], which can lead to EEG misinterpretation and then misdiagnosis. Approximately 20%-30% of patients referred to specialized epilepsy centers are

misdiagnosed [13]. Therefore it is necessary to develop efficient and reliable automatic techniques for ETs detection and classification to help physicians with less experience interpret EEGs.

Methods for automatic detection and classification of ETs have been studied for 40 years since the rise of automatic analysis of EEG in 1970s [19]. Unfortunately, technologies developed so far are still not as reliable as experienced human interpreters. Automated EEG detection is difficult due to several reasons: (1) The morphologies of both ETs and background signals vary widely between patients; and (2) The waveforms of ETs are similar to some normal background activities (i.e. wicket spikes, exaggerated alpha activity, small sharp spikes, and sleep related activities) and also to artifacts (i.e. extracerebral potentials from eye blink, eye movement, muscle, heart, electrode, etc.), which contribute to a large number of false positive detection [26] [21] [22]. Meanwhile, the textbook definitions of ETs supplied by experts are overly simplistic. Development of training and testing datasets to develop ET detection algorithms is also expensive and time consuming due to the insufficient quantity and quality of ET exemplars, which is because obtaining expert opinion from EEG physician experts is expensive and there is disagreement among the experts about the classification of some EEG waveforms [62].

1.2 Previous Related Work

Many approaches aiming to improve the performance of automatic ET detection and classification have been implemented and published since 1970s. Most of them focused on strategies of detection of ETs in the raw EEG signal. Template matching was initially used. It calculates the cross-correlation between a EEG segment and a model ET waveform; and then the decision is made by a pre-selected threshold [57] [18].

Many morphology-based detection strategies were developed later due to their intuitiveness. Gotman et al. [24] interpreted the background context in which a spike occurs and decomposed the waveform by finding segments between amplitude extrema. In order to describe the spike, they introduced the following concepts: (1) the relative height; (2) pseudo-duration (The pseudo-duration is graphically determined by extending a line from the start of a sequence, point A, through the half-way point of the actual EEG wave and extending it so its end, point B, equals the amplitude level of the ending point of the sequence; The horizontal distance from A to B is the pseudo-duration.) of the two half-waves; (3) the relative sharpness at the apex; and (4) the total duration. Guedes

de Oliveira et al. [14] used a normalized standard deviations of the amplitude of the EEG signal; and then they applied a threshold to distinguish spikes from non-spikes. Faure et al. [16] introduced the idea of using duration, amplitude and slope features of half-wave (one side of the triangular shape of a spike). Wilson [62] suggested using background context of a spike to normalize the spike parameters. Wilson [63] used curvature and angles. Many of these algorithms yielded low selectivity since all normal transients, abnormal transients and artifacts fit the same morphologic definition [26].

More sophisticated methods have been proposed. Sankar et al. [52] used an autoregressive model to isolate transients in each 5-second window of EEG and then to classify them as spikes if they match pre-selected templates. Background EEG signals were considered to be stationary in this method. The disadvantage of autoregressive method is that it is sensitive to the number of poles [62].

Features in frequency spectrum were also proposed. Pietila et al. [58] applied an adaptive segmentation on EEG waveform and used the spectral power in a number of frequency bands as features. Other researchers attempted to use spectral analysis methods such as the Fourier Transform [4], the Hilbert Transform [17] and the Walsh transform [3] to interpret EEG signal. These methods have the fixed time-frequency resolution limitation, which means that the increment of resolution in the time domain causes decrease of that in the frequency domain [26].

Wavelet analysis is a relatively new and promising method to extract features [2]. The Wavelet Transform (WT) has the advantage of multiple time/frequency resolution decomposition. Particular characters of signals, such as non-stationary transient events, can be represented in various scales [43]. The WT provides general techniques for ET detection. Senhadji et al. [55] applied the discrete wavelet transform (DWT) to 10-second EEG segments to separate background and artifacts from ET events. For training purpose, Park et al. [48] obtained wavelet coefficients by applying the Daubechies wavelet of order 4 (DB4) on 1-second segments sampled at 256Hz. Goelz et al. [23] applied the continuous wavelet transform (CWT) to generate a detailed spectrum of frequency versus time for background signals; it then searched for events whose spectrum show statistical deviations; these events were considered as ET candidates. Wavelet analysis has been frequently used in recent ET detection methods [19].

The artificial neural network (ANN), a supervised learning machine, has been widely implemented in EEG research, including detection and classification of ETs. Many types of features

have been proposed as input and many structures of ANNs have been developed. Webber et al. [61] used multi-layer perceptron (MLP) networks to test two sets of features. Ozdamar et al. [47] used raw signals as input to a neural net, aiming to seek the best input signal length for MLP. Park et al. [49] used wavelet coefficients in selected subbands as features.

1.3 Purpose of the Research and Overview of the Dissertation

This research intends to make improvements and innovations based on several previous algorithms. Both new and old algorithms are tested on real world rsEEG datasets ¹. Their performances are compared.

As illustrated in Figure 1.1, there are two stages in this research: the detection stage and the classification stage. Both stages process the expert-marked annotations in the EEG recordings, a.k.a. yellow boxes (YBs). The Wavelet Transform is the principal strategy to preprocess the raw data before feature extraction. Both stages adopt features yielded from wavelet subbands. The detection stage also considers a group of pure morphology-based features for comparison.

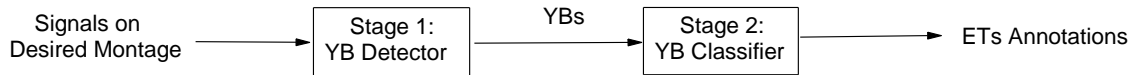


Figure 1.1: The flow chart of the desired ETs detection system

In the detection stage, as illustrated in Figure 1.2, the ET detector identifies a set of candidate events that include abnormal brain activities. Two detection methods are implemented: (1) Apply a threshold; and (2) Train with a neural net. The detection results are compared with a set of yellow-boxes marked by experts.

In the classification stage, as illustrated in Figure 1.3, the classifier goes a step further to determine whether the yellow-box candidates marked by experts are ETs. The major classification algorithm in this research is the k-nearest-neighbor rule. Fuzzy classification attempts are explored

¹<http://eegnet.clemson.edu/>

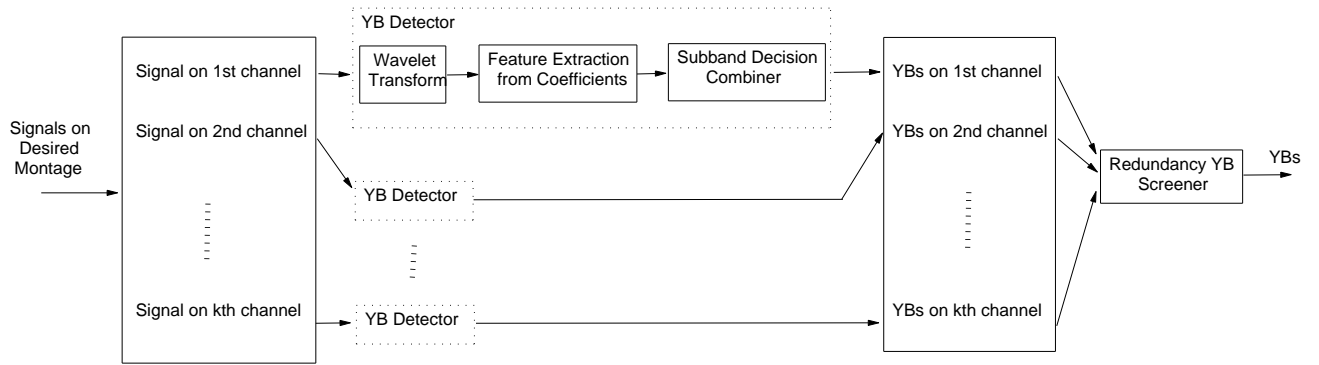


Figure 1.2: The flow chart of the detection module

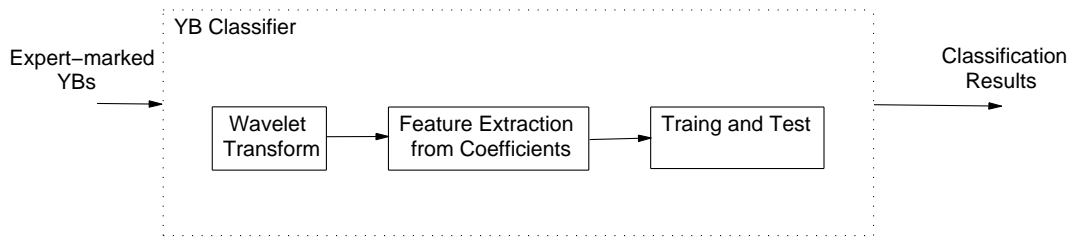


Figure 1.3: The flow chart of the classification module

as well as the crisp strategies. Referential classification results created by experts are compared with the machine classification results.

In this dissertation, Chapter 2 shows the general methodology of electrode placement in EEG recording system, data collection, and wavelet analysis. Chapter 3 shows the specific methodology of classification, including design of the experiments, evaluation method, results and conclusions. Chapter 4 shows the specific methodology of detection, including algorithms adopted by each module of the detectors, results and evaluations. Chapter 5 discusses the implication of the results in Chapter 3 and Chapter 4, and summarizes the dissertation based on the presentations in previous chapters.

Chapter 2

Data Acquisition and Research Methods

2.1 Routine Scalp EEG and the International 10-20 System

An EEG signal is a measurement of currents that flow during synaptic excitation of the dendrites pyramidal neurons in the cerebral cortex. When neurons are activated, the synaptic currents are produced within the dendrites. The summation of the electrical potentials from these dendrites produce an electrical field over the scalp, which can be measured by equipment. The routine scalp EEG recording (rsEEG), the most common type of EEG recording, is mainly used to distinguish epileptic seizures from other brain events. rsEEG is a non-invasive recording and thus is preferred, yet there are disadvantages. The human head consists of several layers: scalp, skull, brain, and other thin layers in between. During the routine EEG recording, the cerebrospinal fluid, skull and scalp will attenuate the EEG signals; moreover, both internal noise from brain and external noise from system is generated in company with the desired signal. Therefore, only electrical potentials generated by a large number of neurons discharging synchronously can generate enough potential that can be recorded by the scalp electrodes [51].

A typical rsEEG lasts for 20-30 minutes. The measurable amplitude range of rsEEG signals is from $10 \mu V$ to $100 \mu V$. A piece of recording is obtained by placing a set of electrodes on the scalp, where conductive gel is applied between electrodes and scalp. Each of the active electrodes

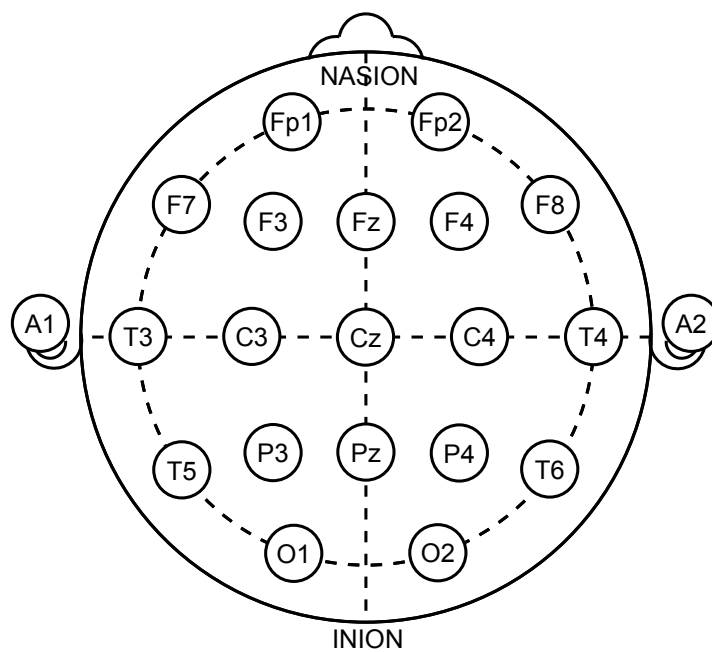


Figure 2.1: 21-electrode International 10-20 system

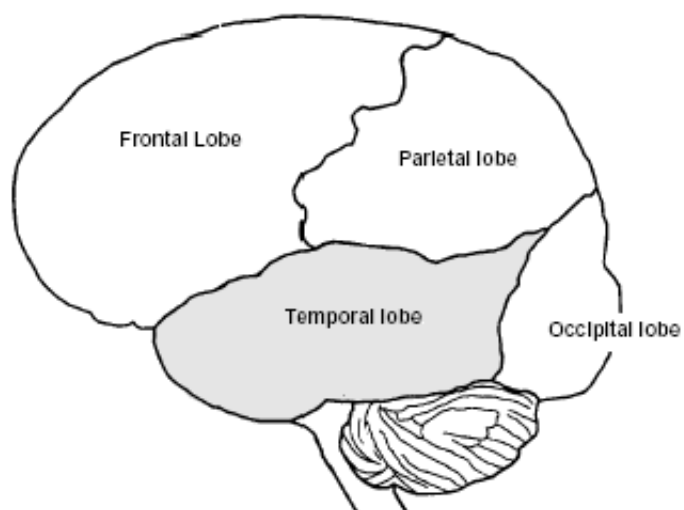


Figure 2.2: Lobes in hemisphere

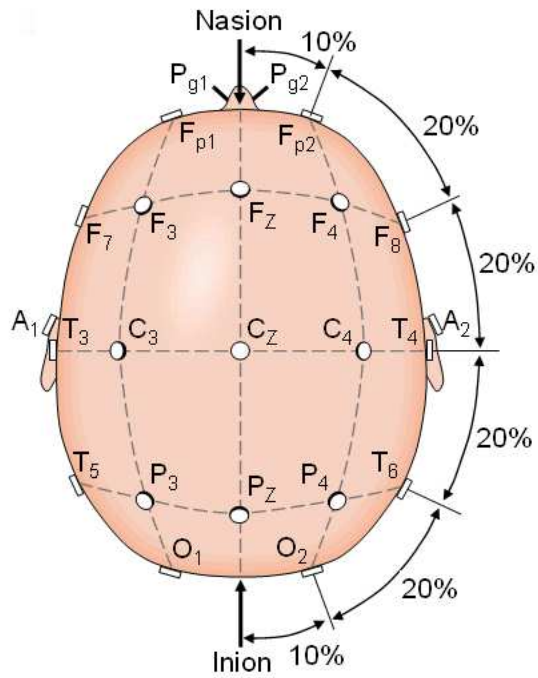
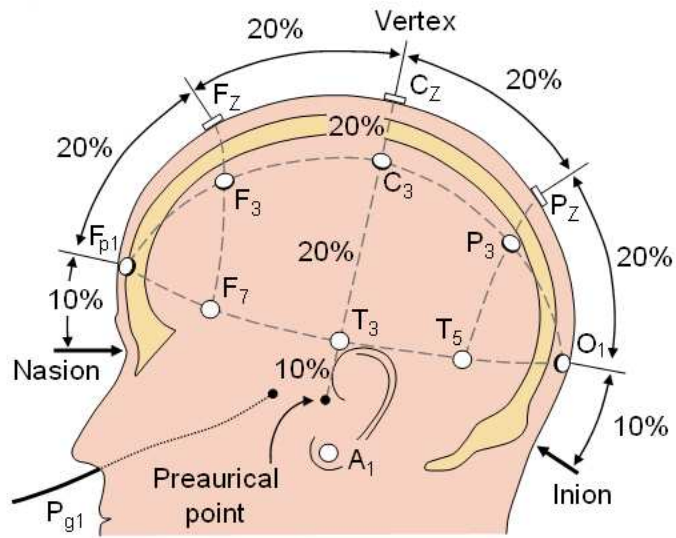


Figure 2.3: Electrode distance

is connected to one input of an individual differential amplifier; a reference electrode is connected to the other input of the amplifier. All the amplifiers are followed by the filter banks, which consist of high-pass filter (usually 0.5 to 1 Hz), low-pass filter (usually 35 to 70 Hz) and notch filter (60

Hz) in routine EEG research. The high-pass and low-pass filters screen out low and high frequency artifacts respectively, while the notch filter removes the noise from electrical power lines. The voltage between the active electrode and the reference is amplified and then filtered. The output is digitized and stored in computerized systems. The effective bandwidth for EEG signals is limited to approximately 100 Hz. Therefore, a minimum sampling rate at 200Hz is often enough to satisfy the Nyquist criterion. Typical sampling rates range from 256 to 512Hz. In some applications, a higher resolution is required for representation of all the brain activities in the frequency domain [51].

Electrode locations are specified by the International 10-20 system for most clinical and research uses. The International 10-20 system is a standardized method to specify the location of the scalp electrodes in EEG recordings for the convenience of comparison between subjects. In most clinical applications, 19 recording electrodes (plus ground and system reference) are required by this standard [56]. In this research, a 21-electrode placement is used as shown in Figure 2.1 ¹. Additional electrodes can be added in between the existing electrodes in the 10-20 system when a higher spatial resolution for a particular area of the brain is required ².

In the standard 10-20 system, certain electrodes are placed to be near certain areas of the cerebral cortex. An electrode location ³ is identified with a letter representing the relevant lobe ('F' - frontal lobe, 'T' - temporal lobe, 'P' - parietal lobe, 'O' - occipital lobe, 'C' - central ⁴) and a number or another letter representing the hemisphere location ('z' refers to the position of electrodes on the midline; even numbers - 2,4,6,8, refer to those on the right hemisphere; and odd numbers - 1,3,5,7, refer to those on the left hemisphere). The phrase '10-20' refers to the fact that the distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance of the skull. Figure 2.3 ⁵ illustrates how the electrodes in a commonly used '10-20' system are arranged using the above rules [45].

Since each channel of an EEG recording is the difference of electrical potential between two electrodes, it can be represented in several formats, which is also referred to as "montage". There are three different types of montages [10]:

1. Bipolar montage: the data in each channel are the differences of the output between two

¹http://upload.wikimedia.org/wikipedia/commons/7/70/21_electrodes_of_International_10-20_system_for_EEG.svg

²<http://www.brainmaster.com/generalinfo/electrodeuse/eegbands/1020/1020.html>

³The location of each lobe on the brain hemisphere is shown in Figure 2.2 from relevant:<http://www.epilepsyfoundation.org/about/types/syndromes/temporallobe.cfm>

⁴'C' for identification of central since there is no central lobe in the cerebral cortex

⁵<http://www.bem.fi/book/13/13.htm#03>

adjacent electrodes; the entire montage consists of the differential data of a series of bipolar electrode pairs;

2. Referential montage: the data in each channel are the differences of the output between an active electrode and a designated reference; and
3. Average reference montage: the data in each channel are the differences of the output between an active electrode and the average reference, which is calculated using the average of the outputs of all the active electrodes.

2.2 Data Acquisition

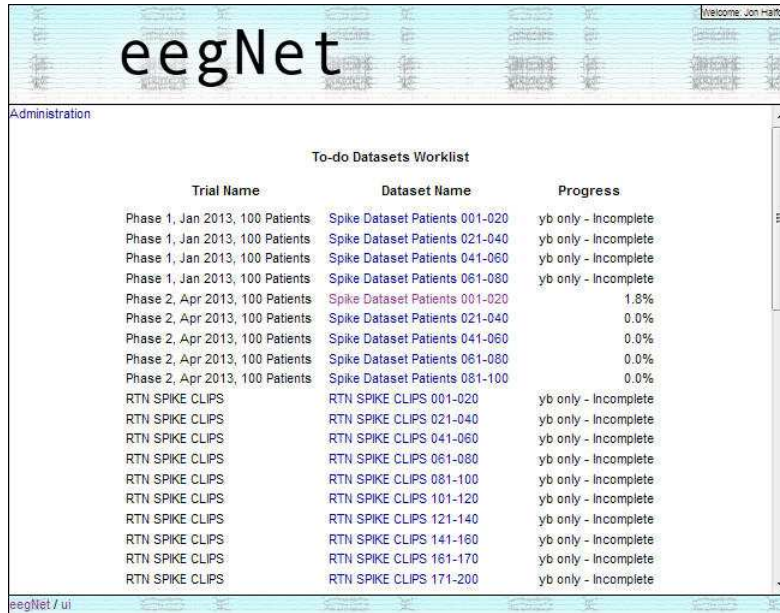
2.2.1 Dataset Based on Crisp Scoring

A selected dataset ('best-7'), created by retrospective review of approximately 1000 rsEEG recordings, was provided by MUSC Neurophysiology Laboratory for clinical purposes and for this research. The dataset contains one hundred 30-second rsEEG segments collected from 100 different rsEEG studies, which were performed on 100 different patients. Fifty of these segments contain ETs from patients with known epilepsy and the other fifty contain benign paroxysmal activity (exaggerated alpha activity, wicket spikes, and small sharp spikes) which can easily be misinterpreted as epileptiform. The rsEEG were recorded referentially (with a digital reference electrode placed between Fz and Cz) at a sampling rate of 256 Hz from 21 channels. The 21 electrodes were placed using the 10-20 system. The EEG data was high-pass filtered (1 Hz), low-pass filtered (70 Hz), and notch filtered (60 Hz). Every twenty 30-second rsEEG segments from all 21 channels were concatenated into a 10-min EEG file. In total, five 10-min EEG files were yielded. The segments with epileptiform activity and those without epileptiform were arranged in a random sequence [27].

The supporting software system of the dataset, EEGNet⁶, is hosted at the School of Computing of Clemson University, as shown in Figure 2.4. EEGnet displays consecutive 10-second rsEEG segment from the 10-min file in a montage at a time with labels on all channel pairs, as shown in Figure 2.5. The software allows users to view the EEG data in several conventional montages, including AP bipolar, transverse bipolar, hatband bipolar, average reference, Cz reference, and ipsilateral ear reference. The users can mark a segment of EEG as an annotation on any channels by placing a

⁶<http://eegnet.clemson.edu>

‘yellow box’ (YB) around it on available montages and classify YB as either (1) an abnormal ET, (2) an artifact, or (3) a burst of electrocortical activity which is not an ET. YBs can be created on multiple montages even if they are representing the same events.



The screenshot shows the EEGNet Administration interface. At the top, there is a header with the EEGNet logo and a 'Welcome, jon@alford' message. Below the header is a section titled 'Administration' containing a 'To-do Datasets Worklist' table. The table has three columns: 'Trial Name', 'Dataset Name', and 'Progress'. The data rows are as follows:

Trial Name	Dataset Name	Progress
Phase 1, Jan 2013, 100 Patients	Spike Dataset Patients 001-020	yb only - Incomplete
Phase 1, Jan 2013, 100 Patients	Spike Dataset Patients 021-040	yb only - Incomplete
Phase 1, Jan 2013, 100 Patients	Spike Dataset Patients 041-060	yb only - Incomplete
Phase 1, Jan 2013, 100 Patients	Spike Dataset Patients 061-080	yb only - Incomplete
Phase 2, Apr 2013, 100 Patients	Spike Dataset Patients 001-020	1.8%
Phase 2, Apr 2013, 100 Patients	Spike Dataset Patients 021-040	0.0%
Phase 2, Apr 2013, 100 Patients	Spike Dataset Patients 041-060	0.0%
Phase 2, Apr 2013, 100 Patients	Spike Dataset Patients 061-080	0.0%
Phase 2, Apr 2013, 100 Patients	Spike Dataset Patients 081-100	0.0%
RTN SPIKE CLIPS	RTN SPIKE CLIPS 001-020	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 021-040	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 041-060	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 061-080	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 081-100	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 101-120	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 121-140	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 141-160	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 161-170	yb only - Incomplete
RTN SPIKE CLIPS	RTN SPIKE CLIPS 171-200	yb only - Incomplete

Figure 2.4: Interface of EEGNet

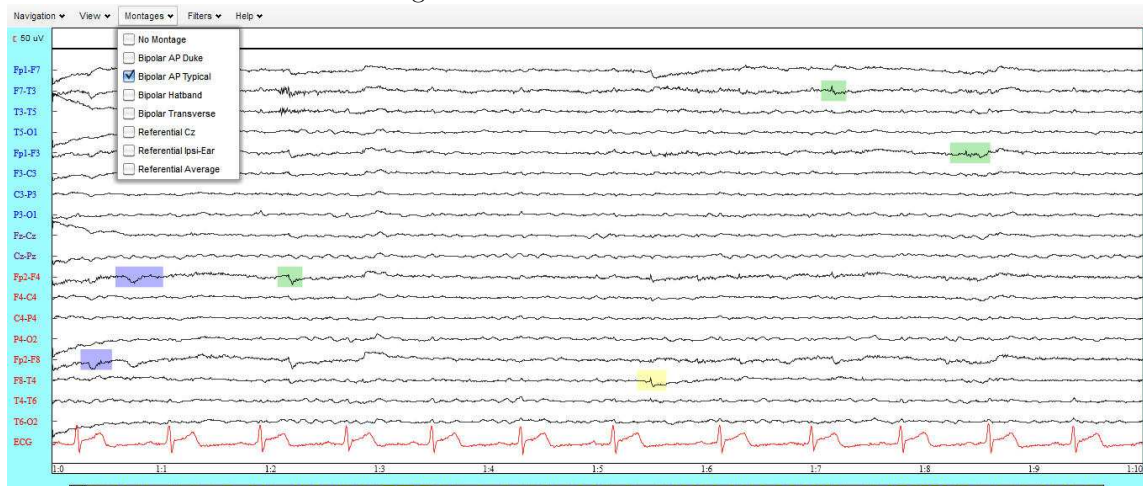


Figure 2.5: 10-second rsEEG data segment in EEGNet

The generation of annotations includes two phases. In the first phase, seven American Board of Clinical Neurophysiology (ABCN) certified academic clinical neurophysiologists (EEGers) were instructed to place YBs around all paroxysmal rsEEG events (PREs, including artifacts, benign electrocortical and epileptiform electrocortical events) in the five 10-min rsEEG files. If several

YBs on more than one rsEEG channel were marked and they were representing the same PRE, the EEGers only kept a single YB that appeared to have the highest amplitude. The redundant YBs were eliminated in three steps: (1) Cluster highly correlated YB candidates which have overlaps in time; (2) Merge two YB candidates into the same cluster if their temporal overlap was at least 50% of the length of the shortest YB candidates; and (3) Choose the YB segment with the maximum sum of correlations to the others in its cluster as the representative YB for the cluster.

In the second phase, eleven EEGers (including the seven from the first phase) were instructed to mark the representative YBs yielding in the first phase as one of the following paroxysmal types:

1. Artifact;
2. Abnormal epileptiform; and
3. Normal electrocortical activity.

Respectively, Figure 2.6 to Figure 2.8 show typical waveforms of the three paroxysmal categories.

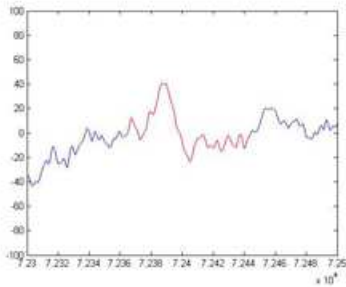


Figure 2.6: Abnormal Epileptiform PED

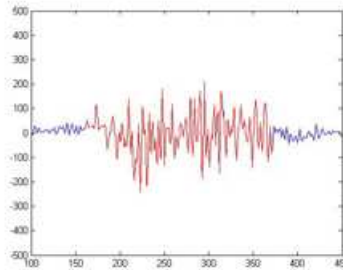


Figure 2.7: Artifact PED

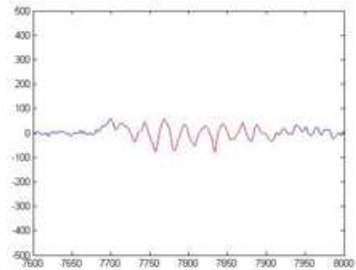


Figure 2.8: Normal Electrocardiac PED

The scoring results of eleven EEGers were output in a file as annotations for this research. The annotations included the following information: annotation ID, start sequence number, end sequence number, channel number, original channel name, notes, scale zoom, thumb nail, paroxysmal type, montage ID, user ID, dataset ID, time stamp, classification ID, trial ID. The most crucial information is listed below:

- annotation ID: identification number of the annotations;
- start sequence number: the sample number at which the annotation starts;
- end sequence number: the sample number at which the annotation ends;

- original channel name: bipolar electrodes information of the annotation;
- paroxysmal type: paroxysmal type of the annotation with default value 'Unclassified'. The “vote” information in an attached file will be used;
- dataset ID: identification number of dataset where the annotation comes from.

The purpose of this research is to distinguish ETs from other EEG events. The 3 paroxysmal types can be merged into two classes:

1. ET class, containing ‘abnormal epileptiform’ paroxysmal type;
2. Non-ET class, containing ‘artifact’ and ‘normal electrocortical activity’ paroxysmal types.

To ensure the credibility of annotations in this study, we only use annotations scored by the seven EEGers with the best inter-rater correlation in the second phase of the generation of annotations. Each YB segment then has seven annotations scored by different EEGers. The only variation of these seven annotations is the paroxysmal type. For research tractability reasons, each YB can only be assigned to one paroxysmal type. The seven EEGers’ opinions about the paroxysmal type on the same event need to be merged. We treated the seven EEGers’ opinions as seven votes and counted votes respectively by paroxysmal type. The paroxysmal type that received most votes from the seven EEGers was recorded in the annotation as the consensus decision for the event.

In total, we derived 83 ETs annotations and 2482 non-ETs annotations from the seven EEGers’ scoring results. They are referred to as ‘best-7’ annotations. The ‘best-7’ annotations are distributed in the following montages:

1. Bipolar AP Typical: F7-T3, T3-T5, P4-O2, T4-T6, Fp1-F7, C3-P3, C4-P4, Fp2-F8, F8-T4, Fz-Cz, T5-O1, P3-O1, Fp1-F3, Cz-Pz, Fp2-F4, T6-O2, F4-C4, F3-C3;
2. Referential Average: F7-avg, C3-avg, Fz-avg, T3-avg, P3-avg, Fp1-avg, A1-avg; and
3. Referential Ipsi-Ear: Fp2-A2, Fp1-A1.

We were also provided with 2998 negative (non-paroxysmal events) annotations for training purpose. A single feature vector was derived from each negative annotation.

Table 2.1: Distribution of the ‘phase2’ confidence factor values by “votes” in paroxysmal types

# experts “vote” AEP	#annotations	average of confidence factors	min of confidence factors	max of confidence factors	average of graded confidence factors
0	1568	0	0	0	0
1	125	0.250666667	0.166667	0.5	1.504
2	53	0.512578616	0.333333	1	1.537736
3	32	0.755208333	0.5	1.333333	1.510417
4	47	1.195035461	0.666667	2	1.792553
5	17	1.480392157	1	2.5	1.776471
6	20	2.066666667	1.166667	3	2.066667
total	1862	0.110275689			1.711111

Table 2.2: Distribution of the ‘phase2a’ confidence factor values by “votes” in paroxysmal types

# experts “vote” AEP	#annotations	average of confidence factors	min of confidence factors	max of confidence factors	average of graded confidence factors
0	911	0	0	0	0
1	136	0.306372549	0.166667	0.5	1.838235294
2	40	0.633333333	0.333333	1.166667	1.9
3	23	0.927536232	0.5	1.333333	1.855072464
4	17	1.490196078	0.666667	2.5	2.235294118
5	25	2.206666667	1.166667	3	2.648
6	16	2.916666667	2.166667	3.666667	2.916666667
total	1168	0.184503425			2.25261324

2.2.2 Dataset Based on Fuzzy Scoring

The annotations discussed in this section for fuzzy classification purpose came from two parts. The first part was derived from another dataset (‘phase2’) supported by EEGnet. This dataset was collected and saved using the same indicator in Section 2.2.1. It also contains 100 patients’ rsEEG recordings (yet different patients from Section 2.2.1). In this dataset, six experts (selected from the eleven experts in Section 2.2.1) were scoring the YBs. Instead of giving a precise opinion about the paroxysmal type of a YB, each expert gave a confidence factor if he believed that the YB contains ETs. The value of the confidence factor is ranging from ONE to FOUR based on the expert’s judgment. FOUR indicates the expert is positive about the fact that the YB contains ETs. ONE indicates it is weakly plausible that the YB contains ETs. If the expert believed there is no ETs in the YB, he marked it as either ‘Artifact’ or ‘Normal electrocortical activity’ class. This annotation set is named ‘phase2’. In total, there are 1862 annotations in this part. The distribution of the confidence factors is listed in Table 2.1

The second part of fuzzy annotation is a set of ‘best-7’ annotations in Section 2.2.1. In this subset, the same six experts used the same rules to score the YBs. They were allowed to leave a YB as ‘Unclassified’ if they believed it does not contain any ETs. This annotation set is named ‘phase2a’ and contains 1168 annotations. The distribution of the confidence factors of ‘phase2a’ is listed in Table 2.2

Confidence factors reflect the membership of being ETs. They determine the feasibility of fuzzy classification.

2.3 Multi-Resolution Analysis and Discrete Wavelet Transform

In Fourier analysis, a segment truncated by a window on original signal is mapped into a one-dimensional sequence of coefficients. The time and frequency resolutions are determined by the fixed width of the analysis window during the entire process. Both time and frequency resolutions are constant. This property makes Fourier analysis only appropriate for periodic signals or for signals with time-invariant statistical characteristics [7]. However, EEG signals are non-stationary. There are several spectral components in EEG signals. From the clinical viewpoint, these components of EEG can be divided into the following bands: delta(0.1 to 3.5 Hz), theta(4 to 7.5 Hz), alpha(8 to 13 Hz), and beta (14 to 30 Hz). From the physiological viewpoint, the most important frequency components are in the range of 0.1 to 30 Hz. EEG signals also contain components referred to as gamma waves, whose frequencies are greater than 30Hz [46]. To decompose these EEG components at different resolution levels, the discrete wavelet transform (DWT) using the strategy of multi-resolution analysis (MRA) is applied.

The MRA analyzes signals at different frequency levels with different resolutions. MRA is designed to give good time resolutions with poor frequency resolutions in high frequency levels and good frequency resolutions with poor time resolutions in low frequency levels. This approach makes sense since in real world, events with high-frequency components usually have short durations and those with low frequency components have long durations.

A multi-resolution representation can be obtained by decomposing the signal using wavelet basis functions. Wavelet means a “small wave” whose windowed function has a finite length (compactly supported). It is used to define a set of basis functions for signal decomposition. It has both

the oscillating characteristic like waves and the ability to allow simultaneous time and frequency analyses. The energy of a wavelet is concentrated in a finite period of time. Wavelet transform is similar to Fourier transform yet much more flexible and informative. It can be made periodic to efficiently represent periodic signals like a Fourier series, moreover, it can be used directly on non-periodic transient signals and yield excellent results. A wavelet expansion maps a one-dimensional signal into a two-dimensional array of coefficients. The two-dimensional representation allows localizing the signal in both time and frequency domains. It is the localizing property of wavelets that is suitable for the analysis of transient, non-stationary or time-varying signal events.

Four properties make wavelet analysis effective [7]:

1. The size of the wavelet expansion coefficients drop off rapidly with expansion level j and most energy of the signal can be represented by a few expansion coefficients;
2. The wavelet expansion allows a more accurate local description and separation of signal characteristics;
3. Wavelets are adjustable and adaptable to fit various applications; and
4. The calculation of DWT only includes multiplications and additions, both of which are basic operations to a digital computer.

DWT analyzes signal at different frequency bands with different resolutions by decomposing the signal into an approximation subband and several detail subbands. DWT employs two closely related sets of functions: scaling function $\varphi(t)$ and wavelet function $\psi(t)$. They are associated with lowpass and highpass filters respectively. The scaling functions and wavelet functions are required to be orthogonal. According to Parseval's theorem, orthogonal basis functions allow a partitioning of the signal energy in the wavelet transform domain. Daubechies showed that it is possible for the scaling function and the wavelet function to have compact support and to be orthonormal, which makes the time localization possible [11] [12]. A basic scaling function is defined as

$$\varphi_k(t) = \varphi(t - k) \quad k \in \mathbf{Z} \quad \varphi \in L^2. \quad (2.1)$$

Define subspace \mathcal{V}_0 as

$$\mathcal{V}_0 = \overline{\text{Span}_k\{\varphi_k(t)\}} \quad k \in \mathbf{Z} \quad \varphi \in L^2. \quad (2.2)$$

A series of scaling functions at different scales can be generated from the basic scaling function by scaling and translation as

$$\varphi_{j,k}(t) = 2^{j/2} \varphi(2^j t - k) \quad k \in \mathbf{Z} \quad (2.3)$$

the span over k is defined as

$$\begin{aligned} \mathcal{V}_j &= \overline{\text{Span}_k \{ \varphi_k(2^j t) \}} \\ &= \overline{\text{Span}_k \{ \varphi_{j,k}(t) \}} \quad k \in \mathbf{Z}. \end{aligned} \quad (2.4)$$

For $j > 0$, $\varphi_{j,k}(t)$ is translated in smaller steps and therefore represents finer detail; for $j < 0$, $\varphi_{j,k}(t)$ is translated in larger steps and represents coarse information. The span is larger for $j > 0$ and smaller for $j < 0$.

The MRA requires the spanned spaces of scaling functions at different levels have a nesting relation as

$$\dots \subset \mathcal{V}_{-2} \subset \mathcal{V}_{-1} \subset \mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \dots \mathcal{V}_j \subset \mathcal{V}_{j+1} \dots \subset L^2 \quad \text{for all } j \in \mathbf{Z} \quad (2.5)$$

with

$$\mathcal{V}_{-\infty} = \{0\}, \quad \mathcal{V}_{\infty} = L^2. \quad (2.6)$$

If $\varphi(t)$ is in \mathcal{V}_0 , it is also in \mathcal{V}_1 , which is spanned by $\varphi(2t)$. Then $\varphi(t)$ can be expressed as

$$\varphi(t) = \sum_n h'(n) \sqrt{2} \varphi(2t - n) \quad n \in \mathbf{Z}. \quad (2.7)$$

The orthogonal complement of \mathcal{V}_j in \mathcal{V}_{j+1} is defined as \mathcal{W}_j . The basis of \mathcal{W}_j is the wavelet functions, defined as $\psi_{j,k}(t)$. $\psi_{j,k}(t)$ span the differences between the various scaling spaces. In general

$$L^2 = \mathcal{V}_0 \oplus \mathcal{W}_0 \oplus \mathcal{W}_1 \oplus \dots \quad (2.8)$$

where \mathcal{V}_0 is the initial space spanned by the scaling function $\varphi(t - k)$. Since $\mathcal{W}_0 \subset \mathcal{V}_1$, $\psi(t)$ can be represented as

$$\psi(t) = \sum_n g'(n) \sqrt{2} \varphi(2t - n) \quad n \in \mathbf{Z}. \quad (2.9)$$

As the scaling functions, wavelet functions at different scales are generated by

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k). \quad (2.10)$$

In MRA, any signal function $f(t) \in L^2(\mathbf{R})$ can be represented by a combination of the scaling functions and wavelet functions as

$$f(t) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(t) + \sum_k \sum_{j=j_0}^{\infty} d_j(k)\psi_{j,k}(t) \quad (2.11)$$

where

$$\varphi_{j_0,k}(t) = 2^{j_0/2}\varphi(2^{j_0}t - k) \quad \psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad (2.12)$$

with coarsest scale j_0 . c_j is the approximation coefficient at scale j and d_j is the detail coefficient at scale j .

In Equation 2.11, the first summation gives a low resolution or coarse approximation of $f(t)$, while each increasing index j in the second summation adds a higher or finer resolution component, which is comparable with the high frequency terms containing signal details in the Fourier series. The coefficients in this wavelet expansion are called the discrete wavelet transform of the signal $f(t)$. In orthogonal system, the approximation coefficients at scale j can be calculated by inner products

$$c_j(k) = \langle f(t), \varphi_{j,k}(t) \rangle = \int f(t)\varphi_{j,k}(t)dt \quad (2.13)$$

and similarly, the detail coefficients at scale j are

$$d_j(k) = \langle f(t), \psi_{j,k}(t) \rangle = \int f(t)\psi_{j,k}(t)dt. \quad (2.14)$$

For multi-stage DWT, Mallat developed a pyramidal algorithm to derive the wavelet coefficients at a lower scale from those at a higher scale [43]. Start from the basic recursion equation

$$\varphi(t) = \sum_n h'(n)\sqrt{2}\varphi(2t - n) \quad (2.15)$$

by scaling and translating the time variable

$$\begin{aligned}\varphi(2^j t - k) &= \sum_n h'(n) \sqrt{2} \varphi(2(2^j t - k) - n) \\ &= \sum_m h'(m - 2k) \sqrt{2} \varphi(2^{j+1} t - m)\end{aligned}\tag{2.16}$$

where $m = 2k + n$. Substitute Equation 2.16 into Equation 2.13

$$c_j(k) = \sum_m h'(m - 2k) \int f(t) 2^{(j+1)/2} \varphi(2^{j+1} t - m) dt\tag{2.17}$$

the integral in Equation 2.17 gives the approximation coefficients at scale $j + 1$. Let $h(n) = h'(-n)$, there is

$$c_j(k) = \sum_m h(2k - m) c_{j+1}(m)\tag{2.18}$$

using Equation 2.14, 2.9, 2.10 and denoting $g(n) = g'(-n)$, we can derive the detail coefficients at scale j as

$$d_j(k) = \sum_m g(2k - m) c_{j+1}(m).\tag{2.19}$$

The Equation 2.18 and 2.19 show that the approximation and detail coefficients at j level can be obtained in two steps: (1) Convolve the scaling coefficients at $j + 1$ level by the time-reversed recursion coefficients $h(n)$ and $g(n)$; then (2) Down-sample by 2 [42] [43]. The scaling function coefficients $h(n)$ and the wavelet function coefficients $g(n)$ are required by orthogonality. They are related by

$$g(n) = (-1)^n h(1 - n)\tag{2.20}$$

for $h(n)$ and $g(n)$ with a finite length N

$$g(n) = (-1)^n h(N - 1 - n).\tag{2.21}$$

The procedure is equivalent to passing the signal through a half band lowpass FIR filter with impulse response $h[n]$ and a highpass FIR filter with impulse response $g[n]$. The original signal is then decomposed into two subbands. The scale of the signal is doubled after down-sampling. Filtering only removes certain frequency components but leaves the scale unchanged. Resolution is a measure of the amount of detail information in the signal and therefore is affected by the filtering

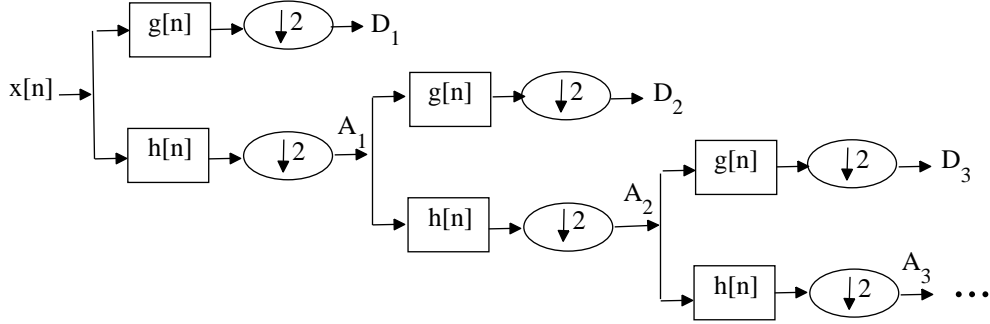


Figure 2.9: Wavelet decomposition tree

operations. The resolution is halved after the filtering operation removing half of the frequency components while the down-sampling operation does not affect the resolution.

In a multi-stage case, the decomposition of the signal can be fulfilled by successive highpass and lowpass filtering of the approximation subband at current scale and decimating the coefficients by 2, as illustrated in Figure 2.9 [43]. The detail and approximation coefficients at the highest scale (level 1 decomposition) are denoted as D_1 and A_1 , respectively. In each level, the index of the coefficients in the successive decompositions will be increased by 1. The frequency resolution is halved after the decomposition in each level. Eventually, the input signal $f(x)$ is decomposed into subbands that correspond to frequency ranges $[0, f_m]$, $[f_m, 2f_m]$, $[2f_m, 2^2 f_m]$, ... $[2^{l-1} f_m, 2^l f_m]$. The frequency ranges of the subbands are directly related to the sampling rate f_s of the input signal, given by

$$f_m = \frac{f_s}{2^{l+1}} \quad (2.22)$$

where l is the level of decomposition. No information has been lost and the original signal can completely be recovered. Those prominent frequency components in the original signal will appear as high amplitude events in the subbands that include part or all of their frequency range. Time localization of these frequencies will also be reflected in the subbands. The time localization also has a resolution depending on which scale these frequencies appear. If the primary information of the signal lies in high frequency range, the time localization of these frequencies will be more precise since they are characterized by more number of coefficients. If the primary information lies in very

low frequency range, the time localization will not be very precise since few coefficients are used to express the signal at these levels. In effect, this procedure offers better time resolution at high frequencies and better frequency resolution at low frequencies. Certain high frequency component can be located better in time domain than a low frequency component; on the contrary, a low frequency component can be located better in frequency domain compared to a high frequency component [50].

Matlab adopts Mallat's algorithm in the calculation of DWT. The DWT of the original signal is obtained by concatenating all coefficients starting from the last level of decomposition. At each level, if the length of the input is N and the length of the filter is $2L$, then the length of the output after downsampling is

$$\text{floor}\left(\frac{N-1}{2}\right) + L$$

which is also the number of wavelet coefficients yielded at current level [29].

2.4 Energy Distribution of EEG Signal by Wavelet Transform

Based on Parseval's theorem, if the scaling and wavelet functions form an orthonormal basis, the energy of the EEG signal can be partitioned at different resolution levels. The energy of the signal $f(t)$ in Equation 2.11 and the energy in each of the components and their wavelet coefficients have the following relation [7]

$$\int |f(t)|^2 dt = \sum_{l=-\infty}^{\infty} |c(l)|^2 + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} |d_j(k)|^2. \quad (2.23)$$

At each decomposition level, the energy of a subband can be presented mathematically as [46]

$$ED_i = \sum_{j=1}^N |D_{ij}|^2, i = 1, 2, \dots, l \quad (2.24)$$

$$EA_l = \sum_{j=1}^N |A_{lj}|^2 \quad (2.25)$$

where $i = 1, 2, \dots, l$ is the wavelet decomposition level. N is the number of coefficients in detail or approximate subband at each decomposition level. ED_i is the energy of the detail subband at decomposition level i , and EA_l is the energy of the approximate subband at decomposition level l .

Chapter 3

Classification of Expert-Marked Yellow-Boxes

3.1 Crisp Classification

3.1.1 Methodology for Design of Classification of Expert-Marked Yellow-Boxes

Wavelet analysis considers the EEG signal as a superposition of different spectra occurring in different time scales at different times and intends to separate them. This procedure is accomplished by the DWT. Two crucial factors that affect the outcome of DWT analysis are: (1) The selection of appropriate mother wavelet; and (2) The number of decomposition levels.

A proper number of decomposition levels is chosen to retain the dominant frequency components of the signal in the wavelet coefficients. In this study, the sampling frequency of the EEG signals is 256 Hz and a 128-sample (500 ms) rectangular window, whose length is long enough to cover a paroxysmal event, is applied on the montages to truncate signals to segments for analysis. Under these circumstances, we choose a 4-level wavelet decomposition. The 128-sample EEG segment was decomposed into 5 subbands (four detail subbands D_1 - D_4 and one approximation subband A_4). Table 3.1 lists the corresponding frequency range of each subband in the 4-level decomposition.

Different mother wavelets are selected for particular applications to achieve maximum ef-

Table 3.1: Corresponding frequency range of each subband in classification

Subband	Frequency Range
D1	64Hz ~ 128Hz
D2	32Hz ~ 64Hz
D3	16Hz ~ 32Hz
D4	8Hz ~ 16Hz
A4	0Hz ~ 8Hz

iciency. In general, DB4 yields the highest correlation coefficients with the epileptic spike among the available wavelet bases in the Matlab toolbox [32], while DB2 possesses smoothing feature that makes it suitable to detect changes of the EEG signals [25]. Figure 3.1 illustrates the similarities between the scaling and wavelet function of DB4 and the shape of an epileptic spike. In this study, six mother wavelets suggested by previous studies are selected and their performances are compared.

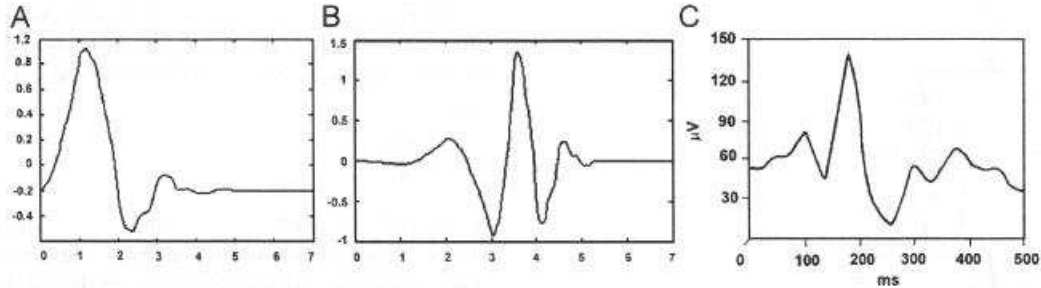


Figure 3.1: Comparison between wavelet functions and epileptic wave form: (A)scaling function of Daubechies 4 (DB4), (B)wavelet function of DB4, and (C)the shape of an epileptic spike [32]

3.1.1.1 Benchmark Wavelet Feature Set

Guler suggested a feature set based on statistics over the wavelet coefficients. First, the signal was truncated using a rectangular window; then the truncated segment is decomposed into 4 levels. Since the WT retains the entire information of the original signal in different subbands, the total length of the wavelet coefficients are no less than that of the segment of the original signal. In this case, a rectangular window with 128 temporal samples is used to obtain EEG segments. Then the signal segment is decomposed into 4 levels, yielding 5 subbands (4 detail subbands D_1 - D_4 and one approximation subband A_4). If the mother wavelet used for decomposition is DB2, as suggested by Guler, there are 65, 34, 18 and 10 wavelet coefficients in the first, second, third and fourth level of detail subband, respectively, and 10 wavelet coefficients in the fourth level of approximation

subband; if the mother wavelet choice is DB4, there are 67, 37, 22 and 14 wavelet coefficients in the first, second, third and fourth level detail subband respectively, and 14 wavelet coefficients in the fourth approximation subband. If all the coefficients are used as input in either case, it will create a high dimension vector with its size over 128.

To reduce the dimension of the feature set, Guler suggested the following statistical features as a substitution [25]:

1. Maximum of the wavelet coefficients in each of the 5 subbands (D_1, D_2, D_3, D_4 and A_4);
2. Minimum of the wavelet coefficients in each of the 5 subbands;
3. Mean of the wavelet coefficients in each of the 5 subbands; and
4. Standard deviation of the wavelet coefficients in each of the 5 subbands

Thereupon, in total we have 20 features in the wavelet-based feature set [25]. This feature set of 20-dimension vectors derived using DB2 is the benchmark of our classification research.

3.1.1.2 Feature Selection

The derivation of the wavelet-based features is an open problem, requiring considerable judgment, computational resources and trial-and-error¹ [5]. Following Guler's methods and elaborating on them, we have developed the following features in each subband:

- **Feature #1:** the highest peak (local maxima) of the wavelet coefficients;
- **Feature #2:** the lowest valley (local minima) of the wavelet coefficients;
- **Feature #3:** the mean of the peaks of the wavelet coefficients;
- **Feature #4:** the mean of the valleys of the wavelet coefficients;
- **Feature #5:** the variance of the peaks and the valleys of the wavelet coefficients;
- **Feature #6:** the variance of the peaks of the wavelet coefficients; and
- **Feature #7:** the variance of the valleys of the wavelet coefficients;

¹This typifies many pattern recognition applications.

In order to achieve high performances with relatively low vector dimensions, we assembled 5 combinations. Their choices of features and dimensions are shown in Table 3.2. Set#1 to Set#5 are basically different combinations of the seven features proposed in the previous paragraph. In our previous study, we found that when using only one of the seven features in the classification, Feature #4 (the mean of the valleys) yields the worst classification result and Feature #5 (the mean of the peaks) yields the second worst result. Thus we discarded the worst in Set#4 and discarded both of them in Set#5.

Table 3.2: Feature choices and dimensions of new feature sets

Selected Features	#1	#2	#3	#4	#5	#6	#7	dimension
Set	Set#1	×	×	×	×	×		25
Set#2	×	×	×	×		×	×	30
Set#3	×	×	×	×	×	×	×	35
Set#4	×	×	×		×			20
Set#5	×	×	×					15

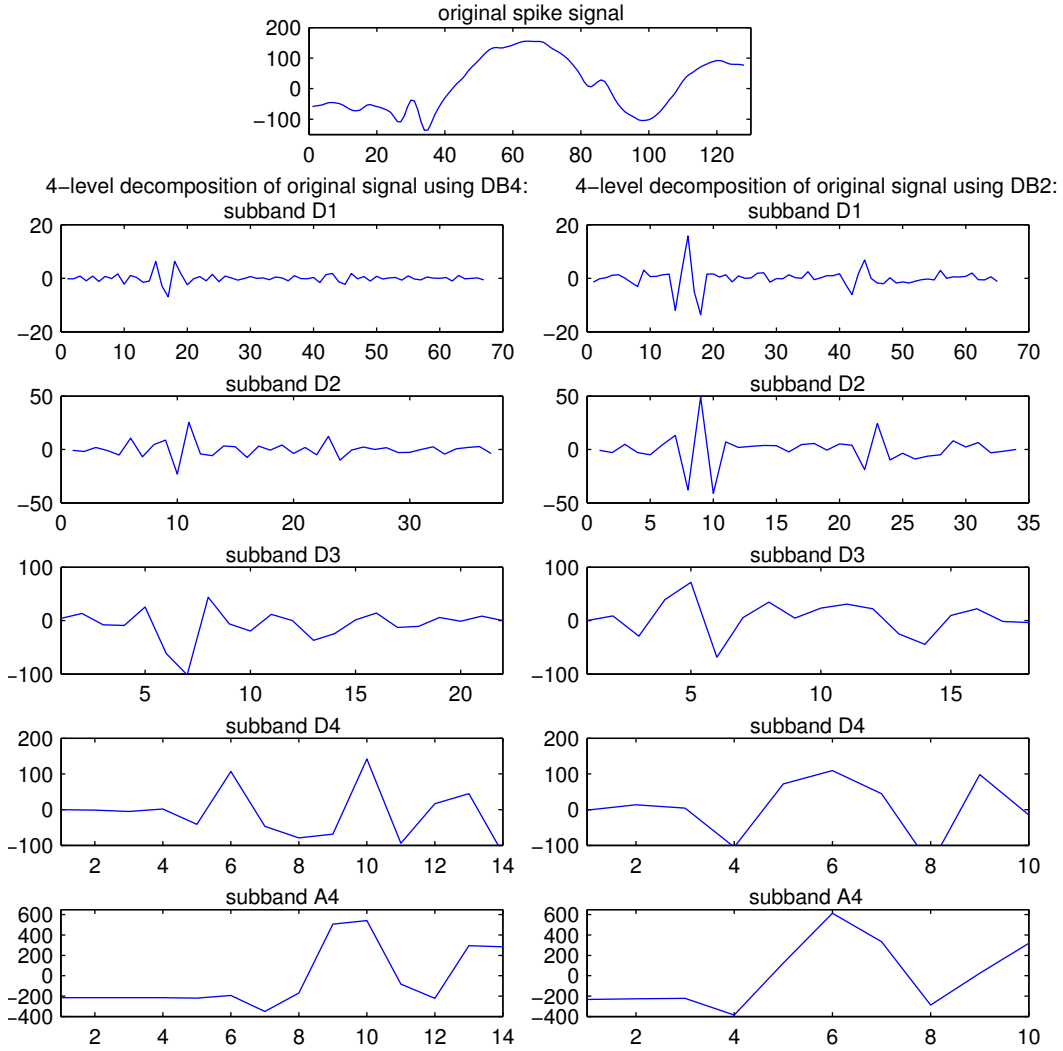
3.1.1.3 Employment of Multiple Mother Wavelets

When using either DB2 or DB4 to decompose, the plots of coefficients of D_1 and D_2 subband indicate signs of respondents to the ET event in the corresponding x-coordinates range where ET occurs in time domain (at x-coordinate 30 in ‘original spike signal’ plot) in Figure 3.2, while the peak values of the respondents using DB2 is twice of that of DB4. This is an example of how a feature can be more evident when implementing WT with different mother wavelets. We have the hypothesis that by combining features yielding from several pre-selected wavelets for the classifier, the performance can be improved. We name this combination ‘multiple mother wavelets strategy’ and we will confirm it in Section 3.1.2. In this research, we combined features yielded by DB4 and DB2. The vector dimension is then doubled when this dual-mother-wavelet strategy is implemented.

3.1.1.4 Scalp Spatial Features

Experts have noticed that the ETs usually occur in the temporal lobe, indicating the spatial information of the electrodes through which the signal is recorded on the scalp could be features. Our previous research showed that attachment of the spatial features to wavelet feature vectors help improve the classification performance in some cases [68].

Figure 3.2: Sample EEG wavelet decomposition results using DB4 and DB2



In this research, we employed a 2D-coordinate system to locate each electrode in the international 10-20 system. The X, Y coordinates of 21 electrodes are computed using the distribution of electrodes described in Figure 2.1 and Figure 2.3. The X, Y coordinates of the midpoint of each bipolar electrode pair are used as the spatial features. The coordinate values are shown in Table 3.3.

3.1.1.5 Methodology of Classification and Performance Evaluation

To test the classification ability of different features and different mother wavelets, 18 datasets are built using the 6 feature sets in Section 3.1.1.2 with 3 choices of mother wavelets:

Table 3.3: Coordinate information of electrode channels

Channel Number	Channel Name	X Coordinate Value	Y Coordinate Value
Channel 1	Fp1	-12.3607	38.0423
Channel 2	F7	-32.3607	23.5114
Channel 3	T3	-40	0
Channel 4	T5	-32.3607	-23.5114
Channel 5	O1	-12.3607	-38.0423
Channel 6	F3	-15.6429	21.6974
Channel 7	C3	-20	0
Channel 8	P3	-15.6429	-21.6974
Channel 9	A1	-50	0
Channel 10	Fz	0	20
Channel 11	Cz	0	0
Channel 12	Fp2	12.3607	38.0423
Channel 13	F8	32.3607	23.5114
Channel 14	T4	40	0
Channel 15	T6	32.3607	-23.5114
Channel 16	O2	12.3607	-38.0423
Channel 17	F4	15.6429	21.6974
Channel 18	C4	20	0
Channel 19	P4	15.6429	-21.6974
Channel 20	A2	50	0
Channel 21	Pz	0	-20

DB2, DB4 and DB4+DB2.

3.1.1.5.1 Balance of the Dataset

In normal EEG recordings, non-ET events occur more frequently than ET events. In our dataset, there are 83 ET feature vectors and 2482 non-ET feature vectors derived from the annotations in total. The ratio of ET/non-ET approximates to 1:30. The annotations also indicated that all 100 patients provided non-ET events while only 31 patients provided ET events. To avoid prejudice in classification, we chose to balance the training set (H); we kept the 1:30 ET/non-ET ratio in the test set S_T to imitate the unbalanced situation in real world.

Within a single trial, 80 ET vectors and 2400 non-ET vectors were randomly selected from the available data. These vectors were divided using the 10-fold cross-validation strategy. The classification is accomplished by implementing the algorithm k-NNR with $k=3$.

3.1.1.5.2 k-Nearest Neighbor Rule

The k-nearest neighbor rule (k-NNR) is a straightforward, non-parametric classification method based on the idea of determining k closest training vectors to the test vector in the feature vector space. It is the simplest machine learning algorithm: a sample is classified by the majority votes of its k nearest neighbors (k is a positive integer, typically a small odd number) [53]. Since k-NNR requires no assumptions about the distribution of the data or the parameters of the classifier, the classification result reflects the properties of the feature data rather than those of the classifiers.

The disadvantage of k-NNR is its high computational complexity, which is extremely time-consuming for large datasets. In this research, there are 2565 feature vectors in total (83 ET vectors and 2482 non-ET vectors). It is a relatively small dataset and k-NNR will satisfy the real-time classification condition.

Ordinary k-NNR measures the Euclidean distance between 2 vectors. In practice, however, the entry values in one vector could be different by several orders of magnitude due to the distribution of the features and the range of the data they represented. To normalize the entry values in a single vector, we computed the distance as following

$$d(\vec{v}_1, \vec{v}_2) = \sqrt{(\vec{v}_1 - \vec{v}_2)^T D^{-1} (\vec{v}_1 - \vec{v}_2)} \quad (3.1)$$

where D is the diagonal of the covariance matrix of the randomly selected single-trial dataset².

3.1.1.5.3 k-Fold Cross-Validation

The size of the training set should be large enough so the classifier can ‘see’ sufficient exemplars. Due to the various morphologies of ETs, it is difficult to determine a reasonable size for a dataset.

A k-fold cross-validation method will satisfy the population of the training set and leave the training and test sets mutually independent. In k-fold cross-validation, the dataset is randomly split into k mutually exclusive subsets, $D_1, D_2 \dots D_k$ of approximately equal size. Then train and test k times, while each time training on $D \setminus D_t$ and testing on D_t with $t = 1, 2, \dots k$ [39]. To evaluate the performance of the k-NNR classifier on small datasets, a stratified k-fold cross-validation is usually used. The folds are stratified so that they contain (approximately) the same proportions of labels

²Preliminary tests indicate using the diagonal of the covariance matrix is superior to using the covariance matrix

as the original dataset [39].

10-fold cross-validation is a recommended method for less bias and variance [39] and thus is used in this research. The dataset is randomly split into 10 mutually exclusive subsets of equal size. Each time the classifier is trained on $D \setminus D_t$ and tested on D_t with $t = 1, 2, \dots, 10$. The overall number of correct classification is used for estimation.

Considering the uncertainty and variation of the random selection of the data in a single trial, 20 trials were performed when using each feature set/wavelet choice. In each trial, the dataset is re-partitioned using 10-fold cross-validation. The mean of the 20 trials is used for evaluation.

3.1.1.5.4 Performance Evaluation

The test performance is assessed by sensitivity and specificity, defined as:

Sensitivity = $TP / (TP + FN)$, capacity to recognize positive events;

Specificity = $TN / (TN + FP)$, capacity to recognize negative activity.

where

TP refers to the data vectors who are classified into the AEP class by both machine and experts;

TN refers to the data vectors who are classified into the nonAEP class by both machine and experts;

FP refers to the data vectors who are classified into the AEP class by machine yet are classified into the nonAEP class by experts;

FN refers to the data vectors who are classified into the nonAEP class by machine yet are classified into the AEP class by experts.

To achieve a single numerical measure that combines sensitivity and specificity, we introduce the measurement of the distance between the result and the coordinate (0,1) in the Receiver Operating Characteristic (ROC) space:

$$distance = \sqrt{(1 - sensitivity)^2 + (1 - specificity)^2}. \quad (3.2)$$

A small distance-to-(0,1) indicates good overall performance. In an ideal case with 100% TP and 0% FP, the distance-to-(0,1) is 0.

3.1.2 Results and Evaluation of Yellow-Box Classification

In this section, to ensure mutual independency of the training set and the test set, 10-fold cross-validation is used to split the dataset; to reduce the effect of the occurrence of outliers, each strategy has been performed 20 times with different data choices, whose average performance is used for evaluation.

3.1.2.1 Comparison of Performances on Selected Feature Set

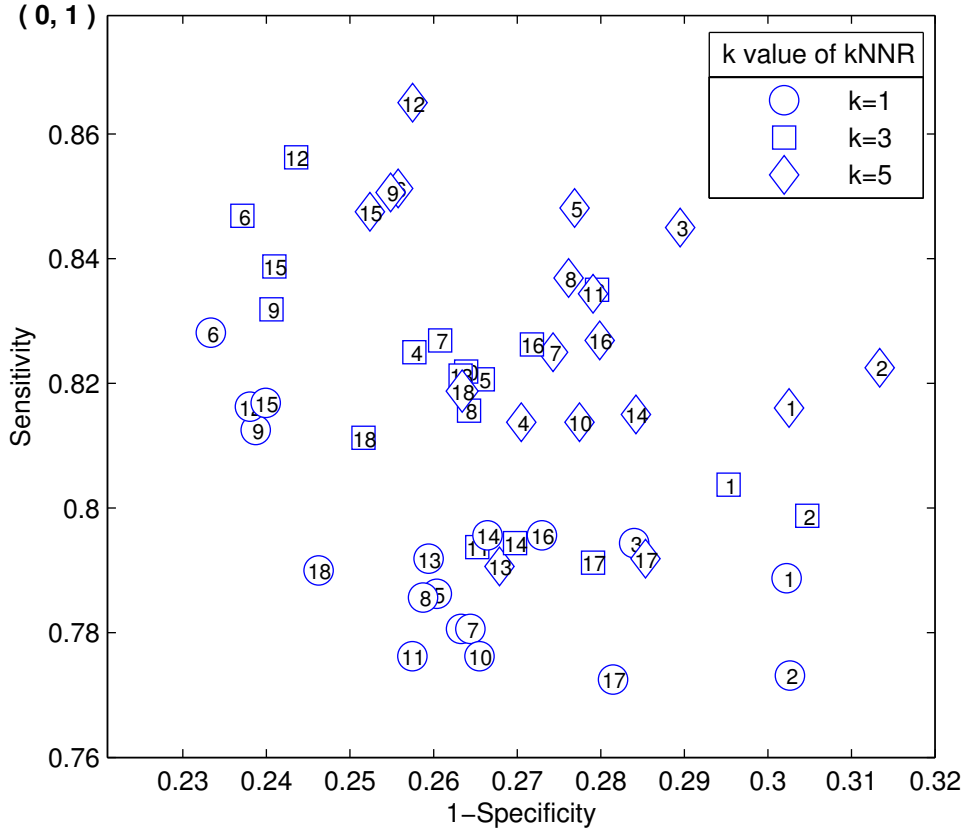
The results of k-NNR ($k = 1, 3, 5$) are summarized in Figure 3.3. When $k = 3$, the average performance of various feature sets are listed in Figure 3.4 and Table 3.4. We considered the benchmark to be Guler’s feature set using this mother wavelet DB2, which showed 79.88% in sensitivity, 69.53% in specificity and 0.3652 in distance. Compared to this benchmark, the sensitivity is improved to 82.06% (+2.18%) and the specificity is improved to 73.41% (+3.88%) by using Set#1 with DB2; the sensitivity is improved to 81.56% (+1.68%) and the specificity is improved to 73.58% (+4.05%) by using Set#2 with DB2; by using dual-wavelet (DB4+DB2) and Guler-features, the sensitivity is improved to 83.50% (+3.62%) and the specificity is improved to 72.05% (+2.52%); assisted by dual-wavelet, Set#1 reached 84.69% (+4.81%) in sensitivity and 76.29% (+6.76%) in specificity while Set#3 reached 85.63% (+5.75%) in sensitivity and 75.64% (+6.11%) in specificity.

Inside each feature set, the sensitivity of our dual-wavelet method is better than that of either single-wavelet method (using either DB4 or DB2) except in Set#5, where the sensitivity of DB4 is the best; the specificity and the distance-to-(0,1) of dual-wavelet are always better than those of either single-wavelet. The specificity of dual-wavelet is more than 2% better than the corresponding result of DB4 (which performed better than DB2), except in Guler Set (+1.58%).

3.1.2.2 Max vs All

In Guler’s method, 5 subbands result from the 4-level wavelet decomposition while 4 features are extracted from each subband. Adding two spatial features, there are 22 features in total. The vector size increases to 42 while using the dual mother wavelet cooperation strategy (20 wavelet derived by DB4 and DB2 respectively plus X & Y coordinates). To reduce the computational

Figure 3.3: Composite summary of feature set evaluations



Mother wavelet / feature set used for each trial (see Section II-A) :

- | | | |
|---------------------|----------------------|----------------------|
| 1. DB4 / Guler | 7. DB4 / Set #2 | 13. DB4 / Set #4 |
| 2. DB2 / Guler | 8. DB2 / Set #2 | 14. DB2 / Set #4 |
| 3. DB4+DB2 / Guler | 9. DB4+DB2 / Set #2 | 15. DB4+DB2 / Set #4 |
| 4. DB4 / Set #1 | 10. DB4 / Set #3 | 16. DB4 / Set #5 |
| 5. DB2 / Set #1 | 11. DB2 / Set #3 | 17. DB2 / Set #5 |
| 6. DB4+DB2 / Set #1 | 12. DB4+DB2 / Set #3 | 18. DB4+DB2 / Set #5 |

Comparative Classification Results of New Feature Sets

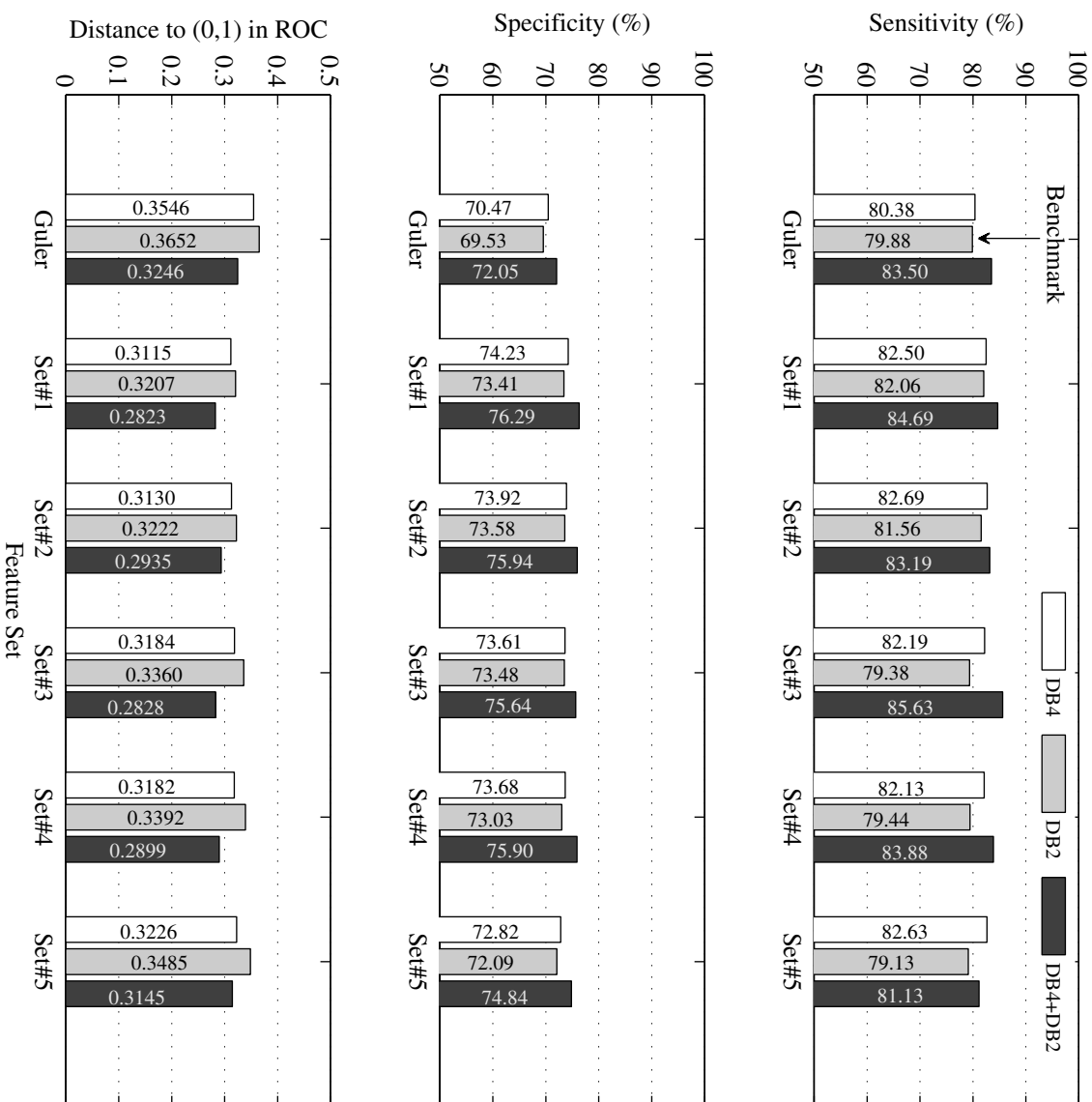


Table 3.4: k-NNR (k=3) comparative classification results of new feature sets

		Sensitivity	Specificity	Distance to (0,1)
Guler	DB4	80.38%	70.47%	0.3546
	DB2	79.88%	69.53%	0.3652
	DB4+DB2	83.50%	72.05%	0.3246
Set#1	DB4	82.50%	74.23%	0.3115
	DB2	82.06%	73.41%	0.3207
	DB4+DB2	84.69%	76.29%	0.2823
Set#2	DB4	82.69%	73.92%	0.3130
	DB2	81.56%	73.58%	0.3222
	DB4+DB2	83.19%	75.94%	0.2935
Set#3	DB4	82.19%	73.61%	0.3184
	DB2	79.38%	73.48%	0.3360
	DB4+DB2	85.63%	75.64%	0.2828
Set#4	DB4	82.13%	73.68%	0.3182
	DB2	79.44%	73.03%	0.3392
	DB4+DB2	83.88%	75.90%	0.2899
Set#5	DB4	82.63%	72.82%	0.3226
	DB2	79.13%	72.09%	0.3485
	DB4+DB2	81.13%	74.84%	0.3145

Table 3.5: k-NNR (k=3) classification results of using overall features based on Guler's features vs. using only maxima

		Sensitivity	Specificity	Distance to (0,1)
DB2	All	79.88%	69.53%	0.3652
	Max	75.00%	68.25%	0.4041
DB4	All	80.38%	70.47%	0.3546
	Max	78.00%	68.14%	0.3872
DB5	All	73.00%	69.98%	0.4037
	Max	72.94%	69.71%	0.4062
DB20	All	76.81%	68.83%	0.3885
	Max	77.13%	67.51%	0.3974
bior1.3	All	76.69%	67.38%	0.4009
	Max	75.13%	68.55%	0.4009
bior1.5	All	77.81%	69.63%	0.3761
	Max	72.00%	66.79%	0.4344
DB4+DB2	All	83.50%	72.05%	0.3246
	Max	77.56%	71.26%	0.3646
DB4+DB2 +bior1.5	All	82.19%	72.13%	0.3308
	Max	76.63%	71.36%	0.3696

Table 3.6: k-NNR (k=3) classification results with/without location features based on Guler’s features

		Sensitivity	Specificity	Distance to (0,1)
DB2	with XY	79.88%	69.53%	0.3652
	no XY	81.31%	67.49%	0.3750
DB4	with XY	80.38%	70.47%	0.3546
	no XY	81.06%	69.53%	0.3588
DB5	with XY	73.00%	69.98%	0.4037
	no XY	77.38%	69.49%	0.3799
DB20	with XY	76.81%	68.83%	0.3885
	no XY	72.50%	69.36%	0.4117
bior1.3	with XY	76.69%	67.38%	0.4009
	no XY	76.25%	67.01%	0.4065
bior1.5	with XY	77.81%	69.63%	0.3761
	no XY	75.50%	68.40%	0.3998
DB4+DB2	with XY	83.50%	72.05%	0.3246
	no XY	82.31%	70.54%	0.3437
DB4+DB2 +bior1.5	with XY	82.19%	72.13%	0.3308
	no XY	78.63%	70.68%	0.3628

Table 3.7: k-NNR (k=3) classification results with datasets of different size

	DS size	Sensitivity	Specificity	Distance to (0,1)
DB2	1240	74.63%	65.69%	0.4268
	1860	78.25%	67.37%	0.3922
	2480	79.88%	69.53%	0.3652
DB4	1240	77.63%	65.78%	0.4088
	1860	78.75%	68.16%	0.3828
	2480	80.38%	70.47%	0.3546
DB5	1240	71.50%	65.71%	0.4459
	1860	73.58%	68.08%	0.4144
	2480	73.00%	69.98%	0.4037
DB20	1240	73.38%	63.65%	0.4506
	1860	77.58%	66.83%	0.4004
	2480	76.81%	68.83%	0.3885
bior1.3	1240	71.25%	63.87%	0.4617
	1860	74.67%	66.44%	0.4205
	2480	76.69%	67.38%	0.4009
bior1.5	1240	73.25%	65.27%	0.4384
	1860	75.08%	67.65%	0.4083
	2480	77.81%	69.63%	0.3761
DB4+DB2	1240	81.13%	67.32%	0.3774
	1860	83.58%	69.83%	0.3434
	2480	83.50%	72.05%	0.3246
DB4+DB2 +bior1.5	1240	75.75%	68.58%	0.3969
	1860	79.92%	70.34%	0.3582
	2480	82.19%	72.13%	0.3308

complexity and to increase the efficiency, we tried using only the maximum of the coefficients in each subband (since a spike usually creates higher coefficients than the background signal at corresponding time). This scheme results in a 7-dimension feature vector. The average results are listed in Table 3.5 by mother wavelet. Note the performance of 22-dimension feature vectors is superior to the 7-dimension case in sensitivity, specificity and distance-to-(0,1), except when using the mother wavelet of DB20 and bior1.3.

3.1.2.3 Effects of Electrode Pair Scalp Location Features

Incorporating the spatial information in the feature vector generally helps to improve classification performance. The average results are listed in Table 3.6 by mother wavelet. By adopting the distance-to-(0,1) in the ROC, we observe that those results with incorporation of the spatial information are closer to the point (0,1), except in the case of mother wavelet of DB5. However, the trend of the changes in sensitivity and specificity shows a more complex situation. The sensitivity improves while the specificity decreases when using the mother wavelet of DB20. The specificity improves while the sensitivity decreases when using the mother wavelet of DB2, DB4 and DB5. Both the sensitivity and the specificity improve when using the mother wavelet of bior1.3, bior1.5 and multiple-wavelet combined feature sets (DB4+DB2 set and DB4+DB2+bior1.5 set). By evaluating the sensitivity only, the best case is that the sensitivity is improved by 4.31% after adding location features when using DB20 feature set. By evaluating the specificity only, the best case is that the specificity is improved by 2.04% after adding location features when using DB2 feature set.

3.1.2.4 Effects of the Size of the Dataset

The dataset used in this study provides a limited number of spike events (83 samples total). It is suggested that increasing the size of the dataset would achieve better results. The effect of the size of the dataset was studied. Three subsets of the available data were used:

1. **2480-set:** 80 ET and 2400 non-ET samples.
2. **1860-set:** 60 ET and 1800 non-ET samples.
3. **1240-set:** 40 ET and 1200 non-ET samples.

The three subsets are formed on the principle that the ratio of ET/non-ET is 1:30, same as in the original dataset. The average classification results are listed in Table 3.7.

Table 3.7 shows that the performance (measurement of distance-to-(0,1) in the ROC) increases with increasing in the size of the dataset. The specificity is definitely improved as the dataset gets larger. However, the sensitivities do not monotonically increase in all cases. The exceptions occur when the features are derived using DB5, DB20 or DB4+DB2, where the sensitivity of 1860-set is the highest in each case respectively. By evaluating the sensitivity only, the best case is that the sensitivity is improved by 6.44% from the 1240-set to the 2480-set when using DB4+DB2+bior1.5 feature set. By evaluating the specificity only, the best case is that the specificity is improved by 4.73% from the 1240-set to the 2480-set when using DB4+DB2 feature set.

3.1.2.5 Statistic Significance of Detection Improvement

3.1.2.5.1 One-Tailed t-Test

To assess statistical significance, a one tailed t-test is used to check whether the mean of the results performed by two different feature sets are statistically different. First, we test if the mean of sensitivities/specificities of a feature set/wavelet combination, is higher than that of benchmark Guler-suggested feature set/wavelet choice, with a significance level α (weakly significant: $\alpha=0.1$; significant level: $\alpha=0.05$; highly significant: $\alpha=0.01$) and 20 observations. The hypotheses are:

$$H_0 : \mu_g = \mu,$$

$$H_1 : \mu_g < \mu,$$

where μ (μ_g) is the mean of the observations of a feature set/wavelet choice (Guler-suggested feature set/wavelet choice) and neither of the variances σ^2 (σ_g^2) of the data is known. The standard deviations of the observations in each set are unequal. With unknown and unequal variances, the t-value is computed by

$$t = \frac{\bar{x}_g - \bar{x}}{\sqrt{s^2/n + s_g^2/n_g}} \quad (3.3)$$

and the degree of freedom of the test is

$$v = \frac{(s^2/n + s_g^2/n_g)^2}{\frac{(s^2/n)^2}{n-1} + \frac{(s_g^2/n_g)^2}{n_g-1}} \quad (3.4)$$

where s (s_g) is the standard deviation of the observations and n (n_g) is the number of observations (20 in our case). The critical region to reject H_0 is $t < -t_\alpha$ [21].

To test if the distance-to-(0,1) decreased significantly, we change the hypotheses:

$$H_0 : \mu_g = \mu,$$

$$H_1 : \mu_g > \mu,$$

and the critical region to reject H_0 is $t > t_\alpha$.

Table 3.8: The highest level at which H_0 can be rejected with different feature set/wavelet choices

Comparison between Different Feature Sets with Same Wavelet Choice								
$x_g \backslash x$	Set#1	Set#2	Set#3	Set#4	Set#5	mother wavelet		
Guler Features	0.05	0.05	0.1	0.1	0.05	DB4	Sensitivity	
	0.05	0.05	fail	fail	fail	DB2		
	fail	fail	0.05	fail	fail	DB4+DB2		
	0.01	0.01	0.01	0.01	0.01	DB4	Specificity	
	0.01	0.01	0.01	0.01	0.01	DB2		
	0.01	0.01	0.01	0.01	0.01	DB4+DB2		
Distance to (0,1)	0.01	0.01	0.01	0.01	0.01	DB4	Distance to (0,1)	
	0.01	0.01	0.01	0.01	0.05	DB2		
	0.01	0.01	0.01	0.01	0.05	DB4+DB2		
Comparison between Benchmark and Different Feature Sets/Wavelet Choice								
Benchmark ³	0.05	0.05	0.05	0.05	0.01	DB4	Sensitivity	
	0.01	0.01	0.01	0.01	0.1	DB4+DB2		
	0.01	0.01	0.01	0.01	0.01	DB4	Specificity	
	0.01	0.01	0.01	0.01	0.01	DB4+DB2		
	Distance to (0,1)	0.01	0.01	0.01	0.01	0.01	DB4	Distance to (0,1)
		0.01	0.01	0.01	0.01	0.01	DB4+DB2	

Table 3.9: The highest level at which H_0 can be rejected of single vs. double mother wavelets

Comparison between Dual-Wavelet and Single-Wavelet within Feature Set									
$x_s \backslash x_d$	Guler	Set#1	Set#2	Set#3	Set#4	Set#5			
DB4							Sensitivity		
	DB4	0.01	0.05	fail	0.01	0.05		fail	
DB2							Sensitivity		
	DB2	0.01	0.01	0.1	0.01	0.01		0.05	
DB4							Specificity		
	DB4	0.01	0.01	0.01	0.01	0.01		0.01	
DB2							Specificity		
	DB2	0.01	0.01	0.01	0.01	0.01		0.01	
DB4							Distance to (0,1)		
	DB4	0.01	0.01	0.01	0.01	0.01		0.1	
DB2							Distance to (0,1)		
	DB2	0.01	0.01	0.01	0.01	0.01		0.01	
Comparison between Dual-Wavelet and Guler's Single-Wavelet									
Guler	DB4		0.01	0.01	0.01	0.01	fail	Sensitivity	
	DB2		0.01	0.01	0.01	0.01	0.1		
	DB4		0.01	0.01	0.01	0.01	0.01	Specificity	
	DB2		0.01	0.01	0.01	0.01	0.01		
	Distance to (0,1)	DB4		0.01	0.01	0.01	0.01	0.01	Distance to (0,1)
		DB2		0.01	0.01	0.01	0.01	0.01	

Table 3.8 and Table 3.9 shows the highest level at which H_0 can be rejected. From Table 3.8, comparing the results that uses the same wavelet choice, we observed: (1) For sensitivity, H_0 is rejected at a significant level ($\alpha = 0.05$) in 2 cases (DB4 and DB2) of Set#1 & Set#2, 1 case (DB4+DB2) of Set#3 and 1 case (DB4) of Set#5; H_0 is rejected at a weakly significant

level ($\alpha = 0.1$) in 1 cases (DB4) of Set#3 & Set#4; (2) For specificity, H_0 is rejected at a highly significant level ($\alpha = 0.01$) in all cases, indicating that the improvement in specificity is both universal and tremendous; (3) Influenced by specificity, H_0 is also rejected at a highly significant level in distance-to-(0,1) except in 2 cases (DB2 and DB4+DB2) of Set#5, where the H_0 is still rejected at a significant level. Comparing the results with benchmark feature set/wavelet choice, we observed: (1) By simply using DB4 instead of DB2,, the sensitivity can be significantly improved and the specificity and distance-to-(0,1) can be highly significantly improved; (2) By employing the dual-wavelet strategy, sensitivities are highly increased; There is an exception in Set#5, where dual-wavelet degrades the sensitivity.

Table 3.9 compares the performances of single-wavelet versus dual-wavelet within feature set. In Table 3.9, we observed H_0 is rejected at a highly significant level ($\alpha = 0.01$) in most cases, especially in half of the sensitivities, indicating that dual-wavelet is a powerful strategy, since Table 3.8 has shown that it is difficult to make improvement in sensitivity. Only two cases failed to reject H_0 at a weakly significant level. Comparing to the benchmark feature set using single-wavelet, all cases of feature sets using dual-wavelet reject H_0 at a highly significant level except the sensitivities of Set#5.

3.1.2.5.2 Power of the Test

The power of a test is the probability of rejecting H_0 given that a specific alternative is true. The power of a test can be computed as $1 - \beta$, where β is the probability of type II error. To find the power at level α , compute the critical region

$$\bar{X}_L = \mu_g + t_\alpha * \frac{\sigma_g}{\sqrt{n_g}} \quad (3.5)$$

the Z-value corresponding to μ when H_1 is true are

$$Z = \frac{\bar{X}_L - \mu}{\sigma/\sqrt{n}} \quad (3.6)$$

and the power of the test [21] is

$$1 - \beta = 1 - P(X < Z). \quad (3.7)$$

Table 3.10: Power with a level of significance of 0.05 (different wavelet choices)

Comparison between Different Feature Sets with Same Wavelet Choice							
$x_g \backslash x_d$	Set#1	Set#2	Set#3	Set#4	Set#5		
Guler	82.19%	83.37%	71.41%	71.14%	85.59%	DB4	Sensitivity
	91.73%	74.40%	0.20%	3.56%	0.88%	DB2	
	21.35%	1.06%	75.30%	5.85%	≈ 0	DB4+DB2	
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4	Specificity
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB2	
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4+DB2	
≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4	Distance to (0,1)	
≈ 1	≈ 1	≈ 1	99.72%	94.99%	DB2		
≈ 1	≈ 1	≈ 1	≈ 1	81.44%	DB4+DB2		
Comparison between Benchmark and Different Feature Sets/Wavelet Choice							
Guler DB2	95.28%	94.49%	91.38%	93.05%	96.39%	DB4	Sensitivity
	≈ 1	99.31%	≈ 1	99.95%	49.81%	DB4+DB2	
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4	
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4+DB2	
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4	Distance to (0,1)
	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	DB4+DB2	

Table 3.11: Power with a level of significance of 0.05 (single vs. double mother wavelets)

Comparison between Dual-Wavelet and Single-Wavelet Inside Feature Set							
$x_s \backslash x_d$	Guler	Set#1	Set#2	Set#3	Set#4	Set#5	
							DB4+DB2
DB4	96.25%	93.52%	8.70%	99.68%	79.30%	≈ 0	Sensitivity
DB2	99.12%	99.61%	73.75%	≈ 1	99.97%	86.58%	
DB4	99.99%	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	Specificity
DB2	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	
DB4	99.98%	≈ 1	98.08%	≈ 1	99.99%	66.42%	Distance to (0,1)
DB2	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	
Comparison between Dual-Wavelet and Guler's Single-Wavelet							
Guler	DB4	≈ 1	96.00%	≈ 1	99.56%	13.62%	Sensitivity
	DB2	≈ 1	99.31%	≈ 1	99.95%	49.81%	
	DB4	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	Specificity
	DB2	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	
	DB4	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	Distance to (0,1)
	DB2	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1	

Table 3.10 and Table 3.11 show the corresponding power of the tests in Table 3.8 and Table 3.9 at a significant level ($\alpha = 0.05$). In an ideal situation, the power should be over 95% when $\alpha = 0.05$. In Table 3.10 and Table 3.11, all power values of specificity tests reach the ideal standard of 95% and all power values of distance-to-(0,1) reach 95% except Set#5/DB4+DB2 vs. Guler/DB4+DB2 in Table 3.10 and Set#5/DB4+DB2 vs. Set#5/DB4 in Table 3.11.

None of the power values of sensitivity tests using the same wavelet choice in Table 3.10 reaches 95%. This result is in expectation, considering the rejection level in Section 3.1.2.5.1. However, when the tests are against benchmark, the power of 6 out of 10 cases reaches 95% and in

another 3 cases it reaches 90%. In Table 3.11, most values are over 95%. When the tests are within feature sets, the power of sensitivity of 6 cases fails to reach 95%. When the tests are against Guler's feature set, the power of sensitivity fails to reach 95% in only 2 cases.

3.2 Fuzzy Classification

In a crisp classification problem, once a data vector is assigned to a class, there is no indication that if it is atypical or representative in that class. In real world, however, many classification problems are based on data that are not fully representative of the class. In order to describe the membership of a data vector in certain class, a fuzzy set is implemented.

Fuzzy set, or class, is characterized by a membership function which associates a data vector with a value in the interval $[0, 1]$, which represents the grade of membership of the data vector in this class. A value of ONE indicates full membership while ZERO means not a member. The nearer the value is to 1, the higher the grade of the membership is in the class. In a crisp case, the membership function takes only two values, 0 and 1. Notice that although there are some resemblances, the membership function is not a probability function. The membership function is nonstatistical in nature. [67] In a fuzzy classification problem, the summation of a vector's membership values in all classes must be 1.0 for mathematical tractability [36].

3.2.1 Fuzzy k-Nearest-Neighbor Algorithm

In the crisp k-nn algorithm, each neighbor is considered equally important when labeling the input vector. The performance is likely to be deteriorated when there are overlaps between the two classes in vector space.

The fuzzy k-nn algorithm assigns a fuzzy membership value in each class to the test vector. It is associated with the membership values of its k nearest neighbors, which will be weighted by their distance (Euclidean, Mahalanobis, etc.) to the test vector in the space.

For a problem with training set $H = \{x_1, x_2, \dots, x_n\}$ and c potential classes, the fuzzy k-nn is accomplished in the following steps [36]:

1. Set $1 \leq k \leq n$;
2. Find the k nearest neighbors by computing and measuring the distance from v_{test} to x_i ;

3. Assign a membership value associated with the i th class to v_{test} as:

$$u_i(v_{test}) = \frac{\sum_{j=1}^k u_{ij} \left(1/\|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^k \left(1/\|x - x_j\|^{2/(m-1)}\right)} \quad (3.8)$$

where u_{ij} is the membership value in the i th class of the j th nearest neighbor and $\sum_{i=1}^c u_{ij} = 1$; the variable m determines how heavily the distance is weighted when calculating each neighbor's contribution to the membership value. As m increases, the neighbors are more evenly weighted. As m approaches to one, the neighbors closer to v_{test} are weighted more heavily. In a conventional case, m is set to two.

3.2.2 Fuzzy c-Means

Clustering is an unsupervised learning strategy. It groups the unlabeled vectors into clusters by maximizing the intraclass similarity and minimizing the interclass similarity, usually through a distance measure. C-means is one of the most widely used clustering algorithms. [64] Following is the fuzzy c-means algorithm [8]:

1. Determine the number of clusters c , $2 \leq c < n$ where n is the number of total data vectors; Determine the constant value m , $1 < m < \infty$; Determine the measurement $\|\mathbf{X} - \mathbf{V}\|^2$;
2. Initialize the membership function U ;
3. At each iteration b , calculate the i th center $v_i^{(b)}$ of the c clusters with $U^{(b)}$; it is expressed in formula

$$v_{il} = \frac{\sum_{k=1}^n (u_{ik})^m x_{kl}}{\sum_{k=1}^n (u_{ik})^m}, \quad l = 1, 2, \dots, p;$$

4. Update the membership $U^{(b)}$ to $U^{(b+1)}$ as follows: For $k = 1, 2, \dots, n$,
 - (a) calculate I_k and \tilde{I}_k :

$$I_k = \{i | 1 \leq i \leq c, d_{ik} = \|x_k - v_i\| = 0\},$$

$$\tilde{I}_k = \{1, 2, \dots, c\} - I_k;$$

- (b) for the k th data vector, compute new membership values as:

i. if $I_k = \phi$,

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}}$$

ii. else $u_{ik} = 0$ for all $i \in \tilde{I}_k$ and $\sum_{i \in I_k} u_{ik} = 1$;

next k ;

5. Compare $U^{(b)}$ and $U^{(b+1)}$ in a convenient matrix norm; if $\|U^{(b)} - U^{(b+1)}\| < \epsilon$, stop; otherwise, set iteration $b = b + 1$, and go to step 3.

3.2.3 Initialization of the Membership Function

There are two crucial issues in fuzzy classification: (1) establishing the ground truth; and (2) de-fuzzification of the outcomes after updating the membership values of the test data. The de-fuzzification can be simply accomplished by applying a threshold on the derived membership values in this specific two-class problem, where the threshold value is set to 0.5 as convention; the test data with membership values above 0.5 are classified as AEP; the test data with membership values below 0.5 are classified as nonAEP. Both issues involve initialization of the membership function. The quality of the membership function will significantly affect the final outcomes. It is worthwhile to note that there are many factors related to the membership function, which leaves the developing of membership function an open issue.

The following sections aim to develop membership functions not only adapting to the dataset but also based on appropriate information. There is no standard procedure to quantify the membership of a given data vector. In a particular case, we are provided with two pieces of information: (1) the confidence factor, and (2) the paroxysmal type ‘voted’ by six experts. Confidence factor is the score that an expert evaluates the likelihood of an event being an ET. It ranges from 0 to 4 while the value 1 to 4 are assigned to the suspected ET events, as mentioned in Section 2.2.2, and the value 0 is assigned to the suspected non-ET events.

An obvious and straightforward way is to adopt the arithmetic mean of the six confidence factors as the membership value. Yet based on the distribution information revealed in Table 2.1 and Table 2.2, we believe the values of the confidence factors in this dataset have been underrated, which is very likely to undermine the performance of classifiers.

To reduce the influence of the underrated confidence factors, the following strategies are

proposed and expected to develop a membership function that can truly represent the quality of the data.

3.2.3.1 Means

3.2.3.1.1 Arithmetic Mean

The membership function values are initialized using the arithmetic mean of the confidence factors. In this case, it is

$$Mem_Val = \frac{1}{6} \sum_{i=1}^6 Confidence_Factor_i. \quad (3.9)$$

Then the membership values are normalized by its superior limit:

$$\overline{Mem_Val} = Mem_Val/4. \quad (3.10)$$

3.2.3.1.2 Geometric Mean

The membership function values are initialized using the geometric mean of the confidence factors. In this case, it is

$$Mem_Val = \sqrt{\frac{1}{6} \sum_{i=1}^6 Confidence_Factor_i^2}. \quad (3.11)$$

The membership values are normalized by Equation 3.10.

3.2.3.1.3 Cube Root of the Cubes' Mean

The membership function values are initialized using the cube root of the mean of the cubes of the confidence factors. In this case, it is

$$Mem_Val = \sqrt[3]{\frac{1}{6} \sum_{i=1}^6 Confidence_Factor_i^3}. \quad (3.12)$$

The membership values are normalized by Equation 3.10.

3.2.3.1.4 Nth Root of the Nth Powers' Mean

The membership function values are initialized using the n th root of the mean of the n th powers of the confidence factors. In this case, it is

$$Mem_Val = \sqrt[n]{\frac{1}{6} \sum_{i=1}^6 Confidence_Factor_i^n}. \quad (3.13)$$

The membership values are normalized by Equation 3.10.

3.2.3.2 Histogram Equalization

Histogram equalization is commonly used to adjust image intensity. It can be used to adjust the probability density function (pdf) of any signal. In an image, when both background and foreground are bright or dark, this technique enhances the global contrast and then highlights the details by spreading the most frequent intensity values.

The algorithm of histogram equalization is straightforward and invertible. It is accomplished in the following steps:

1. Compute the PDF of the dataset

$$pdf[x = i] = \frac{num[x = i]}{n} \quad (3.14)$$

where $num[x = i]$ is the number of occurrences of data i and n is the cardinality of the dataset;

2. Compute the CDF of the dataset

$$cdf[x = i] = \sum_{x=inf\{x\}}^i pdf[x = i]; \quad (3.15)$$

3. Transform into the new values

$$pdf'[x = i] = sup\{x\} * cdf[x = i]. \quad (3.16)$$

In this case, the original confidence factors can take a integer value between zero to four, while the majority of the values are zero. If all the zero values are taken into consideration, the

equalization results will be severely overrated. To avoid overrating, the following five strategies are implemented and the distribution of data before and after equalization is illustrated in Figure 3.5:

1. Equalization based on all individual confidence factors. As mentioned before, due to the large cardinality of the zero values, the new confidence factor values after equalization are pushed to high score zone and are overrated. The distributions before and after equalization are demonstrated in plot '1.all scores' in Figure 3.5;
2. Equalization based on individual confidence factors belonging to the events whose arithmetic means of the six confidence factors are non-zero. If an event receives zero scores from all six experts, then the six scores will be excluded from the calculation, otherwise all of them will be retained. The distributions before and after equalization are demonstrated in plot '2.non-zero-avg-annotation scores' in Figure 3.5;
3. Equalization based on non-zero individual confidence factors. All the zero individual scores are excluded from the calculation, irrespective of the non-zero scores graded by other experts on the same event. The distributions before and after equalization are demonstrated in plot '3.non-zero scores' in Figure 3.5;
4. Equalization based on the average of six experts' confidence factors. First, compute the arithmetic mean of the six confidence factors of each event; then apply the histogram equalization on all the means. The distributions before and after equalization are demonstrated in plot '4.average scores' in Figure 3.5;
5. Equalization based on the non-zeros average of six experts' confidence factors. First, compute the arithmetic mean of the six confidence factors of each event. Second, exclude all the zero mean. Then apply the histogram equalization on the rest means. The distributions before and after equalization are demonstrated in plot '5.non-zero average scores' in Figure 3.5.

Figure 3.6 illustrates the relation between the renewed confidence factor values by equalization and the original values. Based on the information in Figure 3.6, three strategies (equalizing all scores, non-zero-average-annotation scores, and all average scores) severely overrated the confidence factors, leaving only two choices (equalizing non-zero scores and non-zero average scores).

Figure 3.5: Histogram of the 200-patient dataset before and after equalization

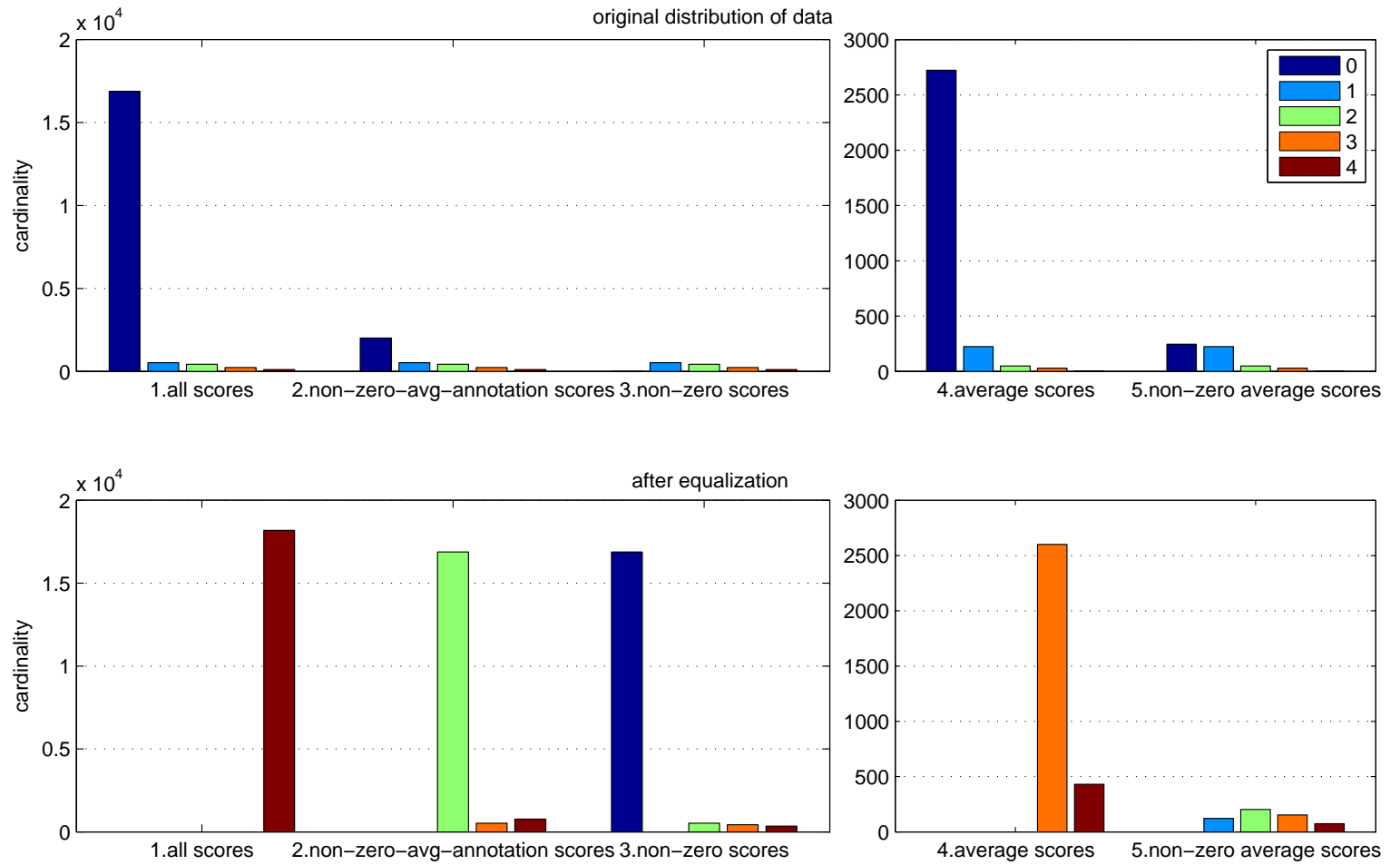
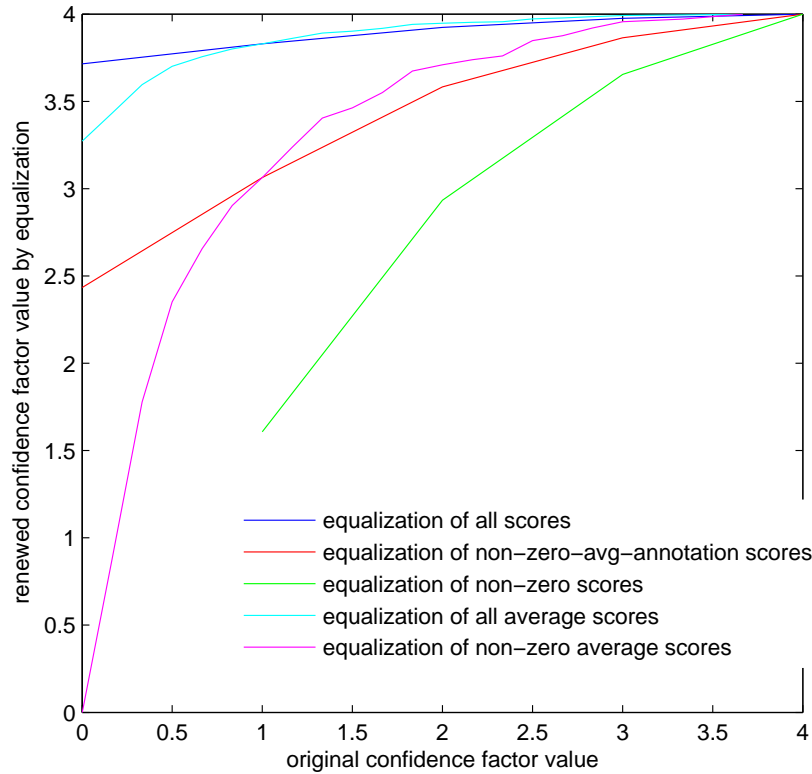


Figure 3.6: Relation between the renewed confidence factor values and their original counterpart



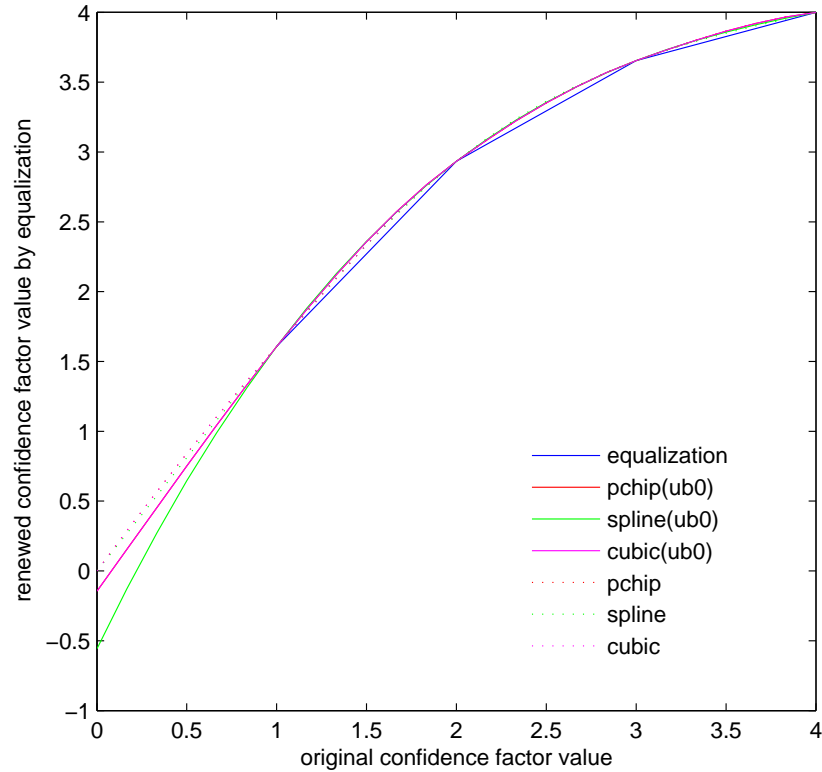
3.2.3.3 Interpolation and Polynomial Fitting

3.2.3.3.1 Interpolation

Among the two selected strategies in the end of Section 3.2.3.2, equalization of non-zero individual confidence factors is a promising method. Since there are six experts scoring each event, their arithmetic mean is a fraction with denominator of 6, and which ranges from zero to four (0, 1/6, 2/6, ... 24/6). Yet in this case, all the confidence factors are integers before equalization. The new values corresponding to these fractions cannot be derived directly by equalization. To estimate the equalization results of the fractions, interpolation is applied to the ‘equalization of non-zero scores’ curve in Figure 3.6.

There are many interpolation strategies. The two most commonly used strategies are spline interpolation and piecewise cubic hermite interpolating polynomial (pchip) interpolation. Spline

Figure 3.7: Spline interpolation and pchip interpolation with determined/undetermined value at zero terminal



interpolation uses low-degree polynomials in each interval and ensures a smooth transition at each knot by specifying the second derivatives. It avoids the problem of Runge’s phenomenon caused by high degree polynomials. Hermite interpolation constructs a function that fits both the given data’s values and their first derivative through the n th derivative. Sometimes the cubic hermite is less smooth [33].

In Figure 3.6, notice that when the equalization is based on non-zero individual confidence factors, the PDF value at zero is vacant. In the following interpolation step, there are two proposals to fill this blank: (1) Fill the vacancy with zero before interpolation (determined value at zero terminal); and (2) Leave the vacancy open, extend the interpolation curve (The ‘equalization of non-zero scores’ curve in Figure 3.6) to zero based on current equalization results from integer point ‘one’ to ‘four’ and the curve will terminate at some value t_0 (undetermined value at zero terminal);

normalize every equalized interpolation result x_i by

$$\overline{x_i} = 4 * \frac{x_i - t_0}{4 - t_0} \quad (3.17)$$

4 is the superior limit of x_i .

Figure 3.7 illustrates the result of spline interpolation and pchip interpolation with either determined or undetermined value at zero terminal. They are extremely close in this case due to the simplicity of the trend of the curve. In order to determine which curve should be adopted in this research, total deviation from the original piecewise equalization curve is measured. Spline is determined as the preferred interpolation strategy. Interpolation with undetermined value at zero terminal is also adopted for it retains the trend of the curve derived from interval [1,4].

3.2.3.3.2 Polynomial Fitting

Interpolation strategy introduces additional data for analysis. However, there is also a concern about its introducing the information of the test data into the training set, which damages the credibility of the outcomes. A remedy is to create a polynomial function that fits the curve instead of direct use of the interpolation result after equalization. This research inspects 2nd to 5th degree of polynomial function by measuring the total deviation from the interpolated curve to the polynomial function. Preliminary tests indicated a 3rd degree polynomial function has the capability to fit the curve with total deviation less than 1e-3, and three is the smallest order that the deviation can be kept less than 1e-3. The 3rd degree polynomial function is then adopted.

3.2.3.4 Function Based Initialization

3.2.3.4.1 Linear Normalization of the “Votes” on the paroxysmal Type

The arithmetic mean also applies to the number of the experts’ opinions on the paroxysmal type of the annotation (a.k.a. “votes”). Six experts in total “voted” on the paroxysmal type of each annotation. The membership function value of ETs class is initialized as:

$$Mem_Val = Votes(AEP)/6. \quad (3.18)$$

3.2.3.4.2 Sigmoid Initialization

Sigmoid function is often used to squash unbounded values in pattern recognition problems, e.g., neural net [53]. Employ a tangent sigmoid transfer function to convert a value x ($x \geq 0$):

$$f(x) = \frac{2}{1 + \exp(-2\alpha x)} - 1 \quad (3.19)$$

where

$$\alpha = -\frac{1}{2x} \ln\left(\frac{2}{f(x) + 1} - 1\right) \quad (3.20)$$

where x can be an expert-assigned confidence factor value or a vote and $f(x)$ corresponds to its converted membership value. We can choose desired confidence factor or vote as classification boundaries and their corresponding α value can be inversely derived from x and $f(x)$. In the relevant tests in Section 3.2.4.1.2, pre-set a boundary x_b and its output y_b ; if $x \geq x_b$, it yields $f(x) \geq y_b$.

3.2.3.4.3 Synthetic Function Based on Confidence Factor and “vote”

Considering a situation that the distribution of the “votes” is inconsistent with that of the confidence factors, a compromise between decision of the “votes” and that of the confidence factors has to be made in the membership function. For instance, membership functions *Mem_Func1* from Equation 3.19 and *Mem_Func2* from Equation 3.18 can be combined with a parameter β :

$$Mem_Fun = \beta * Mem_Func1 + (1 - \beta) * Mem_Func2. \quad (3.21)$$

There is no evidence regarding which decision, “votes” or confidence factors, is more accurate. We choose three β based on experience: (1) 0.5, equal weight; (2) 0.618; and (3) 0.382.

3.2.3.4.4 Multi-Dimension Function Application

In Section 3.2.3.4.3, the two functions respectively derived from “votes” and confidence factor can be considered as a two-dimension function, with the variables “vote” and confidence factor. In fact, the dimension can be expanded to six using the six confidence factors scored by different experts, or even to seven when including the additional variable “vote”. The following functions are implemented:

1. Sigmoid function:

$$f(x) = \frac{2}{1 + \frac{1}{6} \sum_{i=1}^6 \exp(-2\alpha x_i)} - 1 \quad (3.22)$$

where

$$\alpha = -\frac{1}{2x} \ln\left(\frac{2}{f(x) + 1} - 1\right) \quad (3.23)$$

2. Piecewise sigmoid function with fluctuant coefficients:

$$f(x) = \frac{1}{1 + \frac{1}{6} \sum_{i=1}^6 \exp(-2\alpha_k x_i)} \quad (3.24)$$

with

$$\alpha_k = \begin{cases} -\frac{1}{u_1-1} \ln\left(\frac{1}{y_1} - 1\right), & x > Cf \\ -\frac{1}{u_2-1} \ln\left(\frac{1}{y_2} - 1\right), & x \leq Cf \end{cases} \quad (3.25)$$

where u_1 , u_2 are superior and inferior limits of the confidence factors and y_1 , y_2 are their desired outputs respectively; Cf is a predetermined boundary, forcing the confidence factors on either side to adopt different coefficient α_k .

3. polynomial function: the polynomial function related to Section 3.2.3.2 and 3.2.3.3 can also be converted into a multi-dimension case; employ the derived polynomial coefficients and use the original individual confidence factors as input variables of the polynomial function; then compute the arithmetic means of the outputs of the function.

3.2.3.5 Biased Confidence Factor

The significance of the confidence factor's value can vary in different cases. For instance, assume there are two events 'A' and 'B' in our case, where the confidence factors assigned to 'A' by the six experts are [2 2 3 3 3 1] and those to 'B' are [3 0 0 0 0 1]. Obviously, the confidence factor value 3 assigned to 'A' has a higher credibility than it does in 'B' where it deviates from the average. When deriving a membership function value from a set of confidence factors, it is more reasonable to attach heavier weights to those confidence factors with high credibility. The following strategy allows the confidence factors to distinguish their significances autonomously: For a set of confidence factors cf_1, cf_2, \dots, cf_n ($n=6$ in our case), assign a coefficient $b_i, i=1, 2, \dots, n$, to each of them; the

Table 3.12: Customization of the coefficient of a confidence factor based on votes

cf_i \ vote	0	1	2	3	4	5	6
0	b_{00}	b_{01}	b_{02}	b_{03}	b_{04}	b_{05}	b_{06}
1	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}
2	b_{20}	b_{21}	b_{22}	b_{23}	b_{24}	b_{25}	b_{26}
3	b_{30}	b_{31}	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}
4	b_{40}	b_{41}	b_{42}	b_{43}	b_{44}	b_{45}	b_{46}

membership function value is computed by

$$Mem_Val = \frac{\sum_{i=1}^n cf_i * b_i}{\sum_{i=1}^n b_i}. \quad (3.26)$$

When the coefficient b_i is an integer, the strategy can be regarded as replication of significant confidence factors in different degrees.

This strategy can be extended to two-parameter situation by customizing Table 3.12 in this particular case, where we assume both the confidence factor and the “vote” about paroxysmal type have effect on the coefficients.

How to fill the coefficients in Table 3.12 is an open issue. The following methods are considered:

1. Since the ETs is sensitive to both confidence factor and vote, and since there are 6 times 4 possibilities in total, use scale 1 to 24 to fill the table and fill the entries to zero while either “vote” or confidence factor is zero;
2. Stratify the table empirically, fill each part with an integer, starting from 0 or 1;
3. Based on empirical observation, implement appropriate functions to compute a decimal in each entry; Section 3.2.3.4.1 shows the use of linearized “votes” and Section 3.2.3.2 and 3.2.3.3 shows the application of equalization related strategies; it is reasonable to reach the speculation that the coefficients will grow linearly along the row of “vote” and also grow proportional to the equalization curve function along the column of confidence factor; there are two models to imitate this trend:
 - (a) Fill the column b_{i0} or b_{i1} (in this case all b_{i0} are zero) with values on the equalization curve; the rest entries of the table can be filled with the value of their left neighbor plus 1; in the end normalize the table with its maximum entry;

- (b) Fill the column b_{i0} or b_{i1} (in this case all b_{i0} are zero) with values on a curve created by equalization; the range of b_{in} ($n=0, 1$) is from 0 to 4; extend the range to [0 5] and increase all the values proportionally; enter them in column $n + 1$; repeat the extension process in next columns; in the end normalize the table by its maximum entry.

3.2.3.6 Optimization of the Coefficients Using Gradient Descent

Section 3.2.3.5 suggests an algorithm to derive coefficients for the confidence factors based on the knowledge of the distributions of the confidence factors and the “votes”. Thus the coefficients in Table 3.12 are empirical in Section 3.2.3.5. In this section, we explore an approach to optimize the weights by gradient descent, which is commonly used to find a local minimum in pattern recognition.

Assume the six experts score the n th vector of $x_1^n, x_2^n, x_3^n, x_4^n, x_5^n$ and x_6^n . The coefficients assigned to the six scores are from Table 3.12. Then the membership value of AEP class for the n th vector is

$$\begin{aligned}
 mb_{new}^n &= f(x_1^n, x_2^n, x_3^n, x_4^n, x_5^n, x_6^n) \\
 &= \frac{1}{4} \frac{(b_{i1j}^n x_1^n + b_{i2j}^n x_2^n + \dots + b_{i6j}^n x_6^n)}{(b_{i1j}^n + b_{i2j}^n + \dots + b_{i6j}^n)} \\
 &= \frac{1}{4} \frac{\sum_{r=1}^6 b_{i_r j}^n x_r^n}{\sum_{r=1}^6 b_{i_r j}^n}
 \end{aligned} \tag{3.27}$$

where j refers to the “votes” of the n th vector and $i_r = x_r^n$. The derivative $\partial mb_{new}^n / \partial b_{ij}$ is

$$\frac{\partial mb_{new}^n}{\partial b_{ij}} = \frac{1}{4} \frac{(\sum_{i_r=i; vote=j} x_r^n)(\sum_{r=1}^6 b_{i_r j}^n) - (\sum_{i_r=i; vote=j} 1)(\sum_{r=1}^6 b_{i_r j}^n x_r^n)}{(\sum_{r=1}^6 b_{i_r j}^n)^2}. \tag{3.28}$$

Substitute d_j for $\|x - x_j\|^{1/(m-1)}$ in Equation 3.8. After implementing fuzzy k-nearest-neighbor, the membership value of AEP class for the n th vector is

$$\begin{aligned}
 mb_{fuzzy}^n &= \frac{\sum_{p=1}^k u_p (1/d_p^2)}{\sum_{p=1}^k (1/d_p^2)} \\
 &= \frac{\sum_{p=1}^k mb_{new}^p (1/d_p^2)}{\sum_{p=1}^k (1/d_p^2)}.
 \end{aligned} \tag{3.29}$$

The error energy between mb_{new}^n and mb_{fuzzy}^n is

$$\begin{aligned} E^n &= (mb_{fuzzy}^n - mb_{new}^n)^2 \\ &= \left(\frac{\sum_{p=1}^k mb_{new}^p / d_j^2}{\sum_{p=1}^k 1/d_j^2} - mb_{new}^n \right)^2. \end{aligned} \quad (3.30)$$

The derivative of the error energy is

$$\begin{aligned} \frac{\partial E^n}{\partial b_{ij}} &= 2 \left(\frac{\sum_{p=1}^k mb_{new}^p / d_j^2}{\sum_{p=1}^k 1/d_j^2} - mb_{new}^n \right) \left(\frac{\sum_{p=1}^k \frac{1}{d_j^2} \frac{\partial mb_{new}^p}{\partial b_{ij}}}{\sum_{p=1}^k \frac{1}{d_j^2}} - \frac{\partial mb_{new}^n}{\partial b_{ij}} \right) \\ &= 2 (mb_{fuzzy}^n - mb_{new}^n) \left(\frac{\sum_{p=1}^k \frac{1}{d_j^2} \frac{\partial mb_{new}^p}{\partial b_{ij}}}{\sum_{p=1}^k \frac{1}{d_j^2}} - \frac{\partial mb_{new}^n}{\partial b_{ij}} \right) \end{aligned} \quad (3.31)$$

where k is the number of the nearest neighbors. k is not fixed and we can adjust it under multiple circumstances. The total error energy of the test data is

$$E = \sum^n E^n \quad (3.32)$$

and its derivative is

$$\frac{\partial E}{\partial b_{ij}} = \sum^n \frac{\partial E^n}{\partial b_{ij}}. \quad (3.33)$$

The correction of the coefficient is

$$\Delta b_{ij} = -\eta \frac{\partial E}{\partial b_{ij}}. \quad (3.34)$$

In this case, we adopt $\eta = 1/\text{cardinality}(S_T)$, where S_T is the test set. Iterate the correction process and the coefficients will gradually be adjusted to fit the situation. To avoid stopping at a local minimum, a momentum can be added to the correction:

$$\Delta b_{ij}(t) = -\eta \frac{\partial E(t)}{\partial b_{ij}(t)} + \alpha \Delta b_{ij}(t-1). \quad (3.35)$$

It is worthwhile to note that this algorithm intends to achieve optimization by minimizing the error energy between the membership values before and after classification. Neither sensitivity nor specificity is involved because to evaluate these two targets, the algorithm must go through a de-fuzzification process. Gradient descent requires the function to be differentiable so the function must

be continuous. The error energy function fits this condition. By minimizing the error energy between the membership values before and after classification, the system achieves better consistency, which consequently contributes to higher sensitivity and specificity.

3.2.4 Performance on Fuzzy Set

3.2.4.1 Fuzzy k-Nearest-Neighbor

3.2.4.1.1 Benchmark of Crisp k-NNR

To assess the performance of fuzzy k-NNR, a reference yielded by crisp counterpart is necessary. The crisp k-NNR is implemented respectively on dataset ‘phase2’, ‘phase2a’ in Section 2.2.2 and the combined 200-patient dataset. In a classification problem, the first step is to establish the ground truth for the dataset. The ground truth of dataset ‘best-7’ in Section 2.2.1 is established by adopting the paroxysmal type that receives the majority “votes”. The dataset ‘phase2a’ is a subset of ‘best-7’ and could inherit its existing ground truth. However, it does not apply to ‘phase2’ or 200-patient dataset. A new criterion need to be established. A conventional strategy is to apply a threshold on the “votes” of AEP and then to assign data vectors with enough “votes” to AEP class, as illustrated in the upper flowchart in Figure 3.8.

The dataset generated for fuzzy tests includes controversial samples that might degrade the performance of a crisp classifier. A reference performance developed with a better quality dataset will be helpful to observe the influence of the quality of data in experiments. In this research, the quality of dataset is upgraded by discarding annotation samples with relatively small “votes”. This is illustrated in the lower flowchart in Figure 3.8. It is an open issue regarding how many “votes” are enough to guarantee the quality of AEP class. The threshold can vary from one to five “votes”. Preliminary work shows that by setting the threshold at four, the 200-patient dataset can yield the best result. Table 3.13 provides an overview on crisp performance with different datasets and ground truth. The best crisp result yielded using the 200-patient dataset shows 77.03% sensitivity, 70.20% specificity and 0.3763 in distance-to-(0,1), which is also the benchmark for the following fuzzy tests.

3.2.4.1.2 Results of the Fuzzy k-NNR

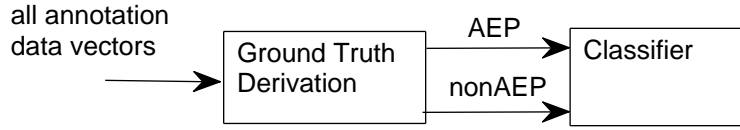
Section 3.2.3 showed multiple possibilities to initialize the membership function. Most strategies have to employ adjustable parameters, which leads to a massive amount of tests. This

Table 3.13: Crisp classification result on 200-patient dataset and selected subset

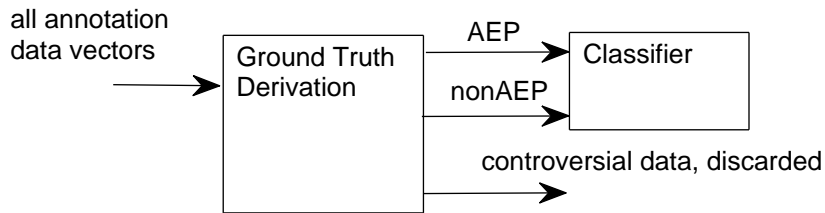
	sensitivity	specificity	dist-(0,1)	$ Ds $
benchmark				
	best7 decision			
best7 annot	84.69%	76.29%	0.2823	2565
phase2a annot	82.30%	71.26%	0.337530288	1168
threshold strategy				
	AEP vote \geq 3			
phase2a annot	82.74%	71.10%	0.3366	1168
phase2 annot	71.81%	68.62%	0.4219	1862
200 annot	74.75%	70.71%	0.3867	3030
	AEP vote \geq 4			
phase2a annot	79.68%	70.32%	0.3597	1168
phase2 annot	72.70%	67.80%	0.4222	1862
200 annot	77.03%	70.20%	0.3763	3030
discard strategy				
	AEP vote \geq 3			
phase2a annot	83.64%	73.77%	0.3091	992
phase2 annot	73.05%	70.92%	0.3965	1684
200 annot	76.45%	73.33%	0.3558	2676
	AEP vote \geq 4			
phase2a annot	81.86%	73.84%	0.3184	969
phase2 annot	75.18%	70.62%	0.3846	1652
200 annot	78.90%	73.46%	0.3390	2621

Figure 3.8: Strategies to determine ground truth for the 200-patient dataset

threshold strategy :



discard strategy :



section only selected some representative tests whose parameter choices help yield relatively decent results. The strategies employed in these selected tests cover the whole list in Section 3.2.3.

As the membership function is created, the ground truth is established by applying a threshold of 0.5, and the data vectors with membership value larger than 0.5 form the AEP class, leaving the rest in nonAEP class. Then the membership values of the test data are computed by fuzzy k-NNR. All the details of the algorithms are described in Section 3.2.1. The classification results are determined by de-fuzzification of the outcome membership values of the test data, where the same threshold criterion to establish the ground truth is used. For a test data vector, if it is assigned to the same class before and after classification based on its original and new membership values respectively, it is a TP/TN; otherwise it is a FP/FN. The same cross-validation and evaluation method is also used here as it is in Section 3.1.2.1.

Table 3.20 shows the results of the fuzzy k-NNR tests. Specific conditions (strategies, parameters, etc.) corresponding to each test are listed as follows:

Condition1: The membership function is initialized using the linear normalization of the “votes” for AEP class of each data vector;

Condition2: The membership function is initialized using the arithmetic mean of the six confidence

factors of each data vector;

Condition3: The membership function is initialized using the geometric mean of the six confidence factors of each data vector;

Condition4: The membership function is initialized using the cube root of the mean of the cubes of the six confidence factors of each data vector;

Condition5: The membership function is initialized using the 4th root of the mean of the 4th powers of the six confidence factors of each data vector;

Condition6: The membership function is initialized using the histogram equalization and interpolation strategy; first, implement the histogram equalization based on all non-zero individual confidence factors; then use spline interpolation to insert values from zero to four with a step length of 1/6; finally, compute the arithmetic mean of the six confidence factors of each data vector and project it to its equalized and interpolated counterpart, which is used as the membership value of this vector. The confidence factor values before and after equalization and interpolation is listed in Table 3.14;

Table 3.14: Confidence factor values before and after equalization and interpolation in Condition6

original	projection	original	projection	original	projection	original	projection
0	0						
0.167	0.376	1.167	2.139	2.167	3.201	3.167	3.764
0.333	0.727	1.333	2.359	2.333	3.324	3.333	3.824
0.500	1.054	1.500	2.561	2.500	3.434	3.500	3.876
0.667	1.358	1.667	2.745	2.667	3.532	3.667	3.923
0.833	1.640	1.833	2.912	2.833	3.619	3.833	3.964
1.000	1.900	2.000	3.064	3.000	3.696	4.000	4.000

Condition7: The membership function is initialized using the polynomial fitting strategy; model the projection function as a 3rd order polynomial (preliminary work has confirmed that an 3rd order polynomial is capable of a perfect fitting in Condition7); adopt the original data and projection data in Table 3.14 as input and output of the polynomial function:

$$projection = \alpha_3 \cdot original^3 + \alpha_2 \cdot original^2 + \alpha_1 \cdot original + \alpha_0. \quad (3.36)$$

Derive the coefficients of the polynomial function and list them in Table 3.15:

the scoring calibration strategy;

Table 3.15: The coefficients of the 3rd order polynomial in Condition7

α_3	α_2	α_1	α_0
0.0339213026	-0.4694708277	2.3351424695	-2.3292694053e-15

Condition8: The membership function is initialized using a tangent sigmoid function defined by Equation 3.19 to transfer “vote” to a membership value; the parameter α computed by Equation 3.20 satisfying the condition that when $vote \geq 4$, the function yields $f(vote) \geq 0.79$;

Condition9: The membership function is initialized using a tangent sigmoid function defined by Equation 3.19 to transfer “vote” to a membership value; the parameter α computed by Equation 3.20 satisfying the condition that when $confidence_factor \geq 0.5$, the function yields $f(confidence_factor) \geq 0.5$;

Condition10: The membership function is initialized using a synthetic function defined by Equation 3.21 with $\beta = 0.382$;

Condition11: The membership function is initialized using a six-variable tangent sigmoid function defined by Equation 3.22; the parameter α computed by Equation 3.20 satisfying the condition that when $confidence_factor \geq 1$ (the minimum value of an individual confidence factor assigned for AEP is one), the function yields $f(confidence_factor) \geq 0.5$;

Condition12: The membership function is initialized using a two-piece sigmoid function defined by Equation 3.24; the indefinite parameter α_k satisfying the condition that when $confidence_factor \geq 4$, the function yields $f(confidence_factor) \geq 0.95$, and when $confidence_factor \leq 0$, the function yields $f(confidence_factor) \leq 0.05$;

Condition13: The membership function is initialized using a six-variable polynomial function whose coefficients are one sixth of their counterpart in Condition7;

Condition14: The membership function is initialized using the biased confidence factor strategy; the membership value of each data vector is derived by Equation 3.26 and the adopted values for coefficient b_{ij} are integers and listed in Table 3.16;

The integer coefficients are created based on the idea of doubling the weights of the larger-than-zero confidence factors;

Table 3.16: The coefficients of vote-based confidence factor in Condition14

cf_i \ vote	0	1	2	3	4	5	6
0	1	1	1	1	1	1	1
1	1	2	2	2	2	2	2
2	1	2	2	2	2	2	2
3	1	2	2	2	2	2	2
4	1	2	2	2	2	2	2

Condition15: The membership function is initialized using the biased confidence factor strategy; the membership value of each data vector is derived by Equation 3.26 and the adopted values for coefficient b_{ij} are integers and listed in Table 3.17;

Table 3.17: The coefficients of vote-based confidence factor in Condition15

cf_i \ vote	0	1	2	3	4	5	6
0	1	1	1	1	1	1	1
1	1	2	2	2	2	3	3
2	1	2	2	2	3	3	3
3	1	2	2	3	3	3	3
4	1	2	2	3	3	3	3

The integer coefficients are selected based on empirical observations;

Condition16: The membership function is initialized using the biased confidence factor strategy; the membership value of each data vector is derived by Equation 3.26 and the adopted values for coefficients b_{ij} are decimals and listed in Table 3.18;

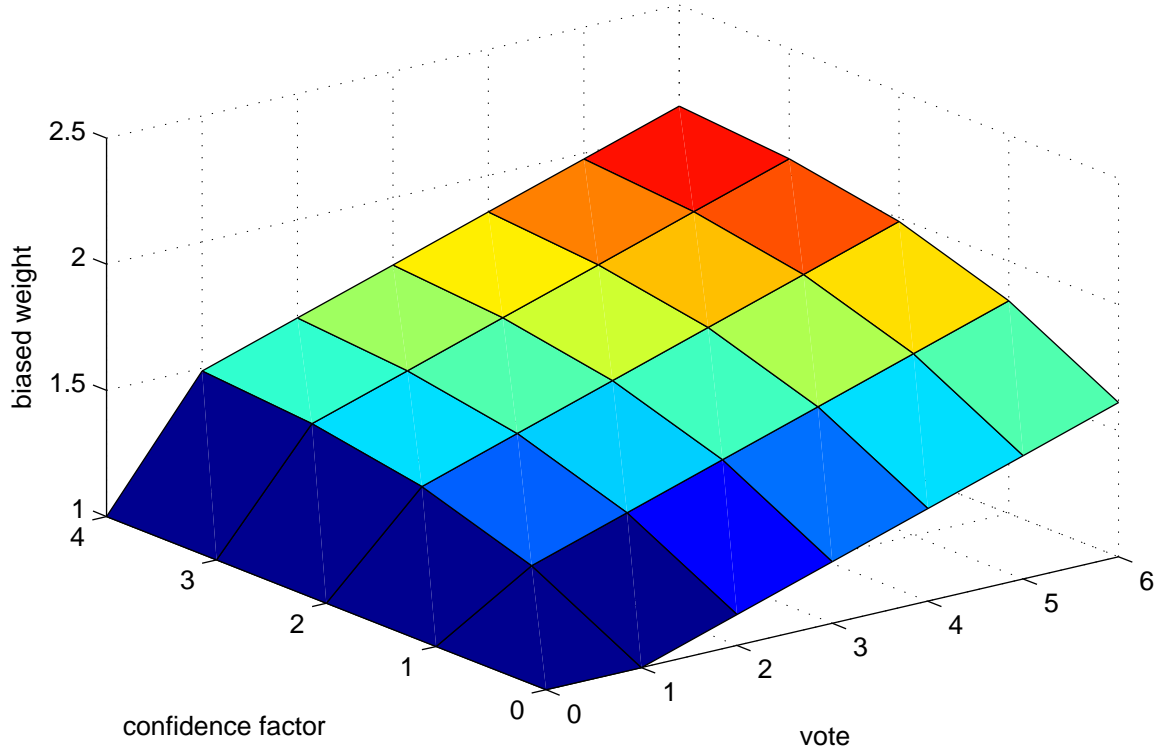
Table 3.18: The coefficients of vote-based confidence factor in Condition16

cf_i \ vote	0	1	2	3	4	5	6
0	1	1	1.122222222	1.244444	1.366666667	1.488889	1.611111
1	1	1.232172	1.354394693	1.476617	1.598839138	1.721061	1.843284
2	1	1.374461	1.49668325	1.618905	1.741127695	1.86335	1.985572
3	1	1.451741	1.573963516	1.696186	1.81840796	1.94063	2.062852
4	1	1.488889	1.611111111	1.733333	1.855555556	1.977778	2.1

The decimal coefficients are created based on the information provided by Equation 3.18 and Section 3.2.3.3; at each “vote” point, the weights increase along the curve created by the polynomial function in Condition7 in interval [0,4]; at each confidence factor point, the weights increase linearly when the number of “vote” goes up; the region of these decimal weights can be adjusted by experiment for optimization purpose; the trend of the biased coefficients is

shown in Figure 3.9.

Figure 3.9: The trend of the biased coefficients in Condition16



Condition17: The membership function is initialized using gradient descent optimization strategy; choose $k = 5$ nearest neighbor when optimizing the coefficients; choose $\eta = 1/\text{cardinality}(S_T)$ and $\alpha = 0.5$ for Equation 3.35; the initial coefficients are all 1; after optimization, the coefficients b_{ij} are updated and listed in Table 3.19:

Table 3.19: The coefficients of gradient descent optimization in Condition17

$cf_i \backslash$ vote	0	1	2	3	4	5	6
0	1	5.735385	1	1.484381	1	0.336674	1
1	1	1.778316	1	0.707368	1	1.163804	1.51228
2	1	0.814442	1	0.545908	1	0.484242	1.660539
3	1	1	1	1	1	0.00074	0.002741
4	1	1	1	1	1	1.821498	0.00552

The membership value of each data vector is then derived by Equation 3.26 using the updated coefficients.

Table 3.20: Results of the fuzzy k-NNR based tests on the 200-patient dataset

neighbor		k=1	k=3	k=5	k=7	k=9		
Condition1	sensitivity	72.82%	67.79%	65.00%	64.57%	63.54%	DB4	
	specificity	66.27%	74.34%	76.73%	77.59%	78.27%		
	distance-to-(0,1)	0.4332	0.4119	0.4203	0.4192	0.4245		
	Condition1	sensitivity	74.50%	70.50%	68.43%	66.93%	65.07%	DB2
		specificity	68.13%	75.50%	77.44%	78.23%	78.80%	
		distance-to-(0,1)	0.4082	0.3835	0.3880	0.3959	0.4086	
	Condition1	sensitivity	74.57%	69.25%	67.04%	66.68%	63.39%	DB4+DB2
		specificity	70.63%	77.90%	80.06%	80.72%	81.18%	
		distance-to-(0,1)	0.3885	0.3787	0.3853	0.3850	0.4116	
Condition2	sensitivity	70.38%	54.88%	53.00%	51.63%	48.25%	DB4	
	specificity	69.22%	87.29%	89.51%	91.00%	91.86%		
	distance-to-(0,1)	0.4272	0.4688	0.4816	0.4921	0.5239		
	Condition2	sensitivity	71.38%	55.13%	52.88%	49.88%	46.00%	DB2
		specificity	69.40%	86.91%	89.09%	90.79%	91.99%	
		distance-to-(0,1)	0.4190	0.4674	0.4837	0.5096	0.5459	
	Condition2	sensitivity	74.13%	54.13%	52.38%	50.00%	48.13%	DB4+DB2
		specificity	71.94%	88.72%	90.84%	92.66%	93.70%	
		distance-to-(0,1)	0.3817	0.4724	0.4850	0.5054	0.5226	
Condition3	sensitivity	72.71%	49.21%	42.79%	39.43%	35.50%	DB4	
	specificity	68.69%	87.13%	90.07%	91.62%	92.74%		
	distance-to-(0,1)	0.4153	0.5239	0.5807	0.6115	0.6491		
	Condition3	sensitivity	76.21%	50.00%	46.21%	42.50%	40.07%	DB2
		specificity	70.17%	87.01%	89.64%	90.84%	91.95%	
		distance-to-(0,1)	0.3815	0.5166	0.5477	0.5823	0.6047	
	Condition3	sensitivity	74.21%	50.93%	43.64%	40.00%	38.07%	DB4+DB2
		specificity	72.36%	88.73%	91.32%	92.69%	93.59%	

	distance-to-(0,1)	0.3780	0.5035	0.5702	0.6044	0.6226	
Condition4	sensitivity	71.72%	53.06%	48.50%	45.67%	42.50%	DB4
	specificity	68.69%	84.89%	87.56%	89.05%	90.39%	
	distance-to-(0,1)	0.4219	0.4932	0.5298	0.5543	0.5830	
	sensitivity	75.67%	55.67%	52.67%	50.11%	45.39%	DB2
	specificity	70.37%	85.02%	87.48%	88.58%	89.60%	
	distance-to-(0,1)	0.3834	0.4680	0.4896	0.5118	0.5559	
	sensitivity	76.44%	55.50%	50.94%	46.50%	43.67%	DB4+DB2
	specificity	73.75%	87.28%	89.43%	90.55%	91.79%	
	distance-to-(0,1)	0.3527	0.4628	0.5018	0.5433	0.5693	
Condition5	sensitivity	69.19%	49.92%	43.35%	40.08%	35.96%	DB4
	specificity	66.79%	84.94%	88.26%	90.04%	91.25%	
	distance-to-(0,1)	0.4530	0.5229	0.5786	0.6074	0.6463	
	sensitivity	76.12%	52.62%	45.88%	42.35%	38.23%	DB2
	specificity	69.63%	85.77%	88.24%	89.78%	90.87%	
	distance-to-(0,1)	0.3863	0.4948	0.5538	0.5855	0.6244	
	sensitivity	72.19%	51.27%	44.12%	40.69%	36.15%	DB4+DB2
	specificity	71.48%	87.94%	90.11%	91.42%	92.74%	
	distance-to-(0,1)	0.3983	0.5020	0.5675	0.5993	0.6426	
Condition6	sensitivity	73.67%	57.79%	53.13%	51.13%	47.67%	DB4
	specificity	67.89%	81.99%	84.23%	85.90%	87.05%	
	distance-to-(0,1)	0.4153	0.4589	0.4946	0.5087	0.5391	
	sensitivity	76.21%	60.46%	54.63%	53.21%	49.92%	DB2
	specificity	69.79%	82.35%	84.08%	85.33%	86.45%	
	distance-to-(0,1)	0.3846	0.4330	0.4809	0.4904	0.5188	
	sensitivity	75.63%	58.08%	55.25%	51.83%	47.79%	DB4+DB2
	specificity	72.42%	85.29%	87.09%	88.18%	89.31%	
	distance-to-(0,1)	0.3681	0.4442	0.4657	0.4960	0.5329	
	sensitivity	72.54%	54.96%	50.63%	47.96%	45.33%	DB4
	specificity	67.44%	81.64%	84.36%	85.94%	87.13%	

Condition7	distance-to-(0,1)	0.4259	0.4864	0.5179	0.5391	0.5616	
	sensitivity	77.50%	60.67%	56.63%	53.46%	51.00%	DB2
	specificity	70.01%	82.99%	84.96%	86.13%	87.15%	
	distance-to-(0,1)	0.3749	0.4285	0.4591	0.4857	0.5066	
	sensitivity	75.71%	58.88%	56.21%	51.04%	46.75%	DB4+DB2
	specificity	72.42%	85.33%	87.12%	88.25%	89.36%	
	distance-to-(0,1)	0.3675	0.4366	0.4565	0.5035	0.5430	
Condition8	sensitivity	72.66%	67.50%	64.11%	63.34%	62.61%	DB4
	specificity	67.25%	74.15%	77.16%	77.94%	78.41%	
	distance-to-(0,1)	0.4266	0.4153	0.4254	0.4279	0.4318	
	sensitivity	72.32%	68.03%	65.87%	65.08%	64.63%	DB2
	specificity	67.66%	75.15%	78.05%	78.58%	79.00%	
		distance-to-(0,1)	0.4257	0.4049	0.4058	0.4097	0.4113
Condition9	sensitivity	72.37%	69.21%	66.39%	64.71%	63.76%	DB4+DB2
	specificity	69.92%	77.36%	80.52%	80.90%	81.31%	
	distance-to-(0,1)	0.4085	0.3822	0.3884	0.4013	0.4077	
	sensitivity	71.38%	57.85%	56.43%	54.82%	53.83%	DB4
	specificity	66.78%	79.82%	82.25%	83.44%	84.10%	
		distance-to-(0,1)	0.4385	0.4673	0.4704	0.4812	0.4883
Condition10	sensitivity	71.78%	60.68%	57.95%	56.20%	55.92%	DB2
	specificity	67.88%	80.83%	83.09%	83.93%	84.68%	
	distance-to-(0,1)	0.4276	0.4374	0.4532	0.4665	0.4667	
	sensitivity	71.62%	58.58%	56.47%	53.80%	53.00%	DB4+DB2
	specificity	69.37%	82.61%	85.16%	86.07%	86.70%	
		distance-to-(0,1)	0.4176	0.4492	0.4599	0.4825	0.4885
Condition10	sensitivity	72.18%	62.10%	60.00%	59.58%	57.65%	DB4
	specificity	67.35%	78.08%	79.85%	80.96%	81.85%	
	distance-to-(0,1)	0.4289	0.4378	0.4479	0.4468	0.4608	
	sensitivity	73.25%	61.73%	61.33%	60.40%	59.35%	DB2
specificity	67.99%	79.17%	81.00%	81.92%	82.49%		

	distance-to-(0,1)	0.4171	0.4357	0.4309	0.4353	0.4426	
	sensitivity	72.95%	62.65%	60.83%	59.83%	57.63%	
	specificity	69.99%	81.30%	83.19%	84.20%	84.82%	DB4+DB2
	distance-to-(0,1)	0.4040	0.4177	0.4263	0.4317	0.4501	
Condition11	sensitivity	70.86%	65.43%	62.57%	62.57%	61.21%	
	specificity	64.36%	70.96%	74.22%	75.39%	76.29%	DB4
	distance-to-(0,1)	0.4603	0.4515	0.4545	0.4479	0.4546	
	sensitivity	74.43%	69.00%	64.86%	64.07%	63.36%	
	specificity	65.93%	72.20%	74.39%	75.40%	76.08%	DB2
	distance-to-(0,1)	0.4260	0.4164	0.4348	0.4354	0.4376	
Condition12	sensitivity	75.57%	66.86%	63.71%	61.21%	60.86%	
	specificity	67.57%	74.45%	76.89%	77.95%	78.76%	DB4+DB2
	distance-to-(0,1)	0.4060	0.4185	0.4302	0.4462	0.4454	
	sensitivity	67.83%	49.50%	41.83%	41.00%	39.17%	
	specificity	61.52%	77.13%	80.43%	82.15%	83.56%	DB4
	distance-to-(0,1)	0.5015	0.5544	0.6137	0.6164	0.6302	
Condition13	sensitivity	66.17%	49.83%	44.00%	40.00%	34.50%	
	specificity	60.73%	76.61%	80.03%	82.79%	84.91%	DB2
	distance-to-(0,1)	0.5183	0.5535	0.5945	0.6242	0.6722	
	sensitivity	69.83%	52.17%	47.17%	44.83%	42.50%	
	specificity	64.31%	78.38%	81.37%	83.33%	84.83%	DB4+DB2
	distance-to-(0,1)	0.4673	0.5249	0.5602	0.5763	0.5947	
Condition13	sensitivity	70.25%	50.69%	47.75%	45.00%	42.88%	
	specificity	66.69%	83.20%	85.66%	87.29%	88.65%	DB4
	distance-to-(0,1)	0.4466	0.5210	0.5418	0.5645	0.5824	
	sensitivity	76.56%	54.81%	50.56%	47.56%	45.31%	
	specificity	68.79%	83.57%	85.20%	86.65%	87.93%	DB2
	distance-to-(0,1)	0.3903	0.4808	0.5161	0.5411	0.5600	
Condition13	sensitivity	73.69%	51.44%	49.19%	45.50%	43.13%	
	specificity	70.59%	85.97%	87.65%	89.59%	91.10%	DB4+DB2

	distance-to-(0,1)	0.3946	0.5055	0.5229	0.5549	0.5757	
Condition14	sensitivity	71.92%	48.00%	43.17%	40.42%	38.67%	DB4
	specificity	69.56%	87.84%	91.07%	92.61%	93.57%	
	distance-to-(0,1)	0.4142	0.5340	0.5753	0.6004	0.6167	
	sensitivity	74.25%	48.25%	45.50%	41.83%	39.67%	DB2
	specificity	70.66%	88.05%	90.69%	92.26%	93.41%	
	distance-to-(0,1)	0.3904	0.5311	0.5529	0.5868	0.6069	
	sensitivity	76.67%	49.75%	43.17%	42.42%	40.17%	DB4+DB2
specificity	73.18%	89.56%	92.31%	93.79%	94.65%		
distance-to-(0,1)	0.3555	0.5132	0.5735	0.5792	0.6007		
Condition15	sensitivity	71.43%	46.86%	40.57%	37.79%	34.00%	DB4
	specificity	69.57%	87.47%	91.10%	92.72%	94.00%	
	distance-to-(0,1)	0.4174	0.5460	0.6009	0.6264	0.6627	
	sensitivity	72.57%	48.36%	42.14%	37.79%	32.79%	DB2
	specificity	70.33%	87.55%	90.50%	92.06%	93.43%	
	distance-to-(0,1)	0.4041	0.5312	0.5863	0.6272	0.6753	
	sensitivity	76.93%	51.36%	41.36%	37.14%	34.71%	DB4+DB2
specificity	73.71%	89.54%	92.44%	94.20%	95.07%		
distance-to-(0,1)	0.3498	0.4975	0.5913	0.6312	0.6547		
Condition16	sensitivity	72.20%	50.60%	46.70%	45.20%	41.40%	DB4
	specificity	68.59%	87.07%	89.97%	91.40%	92.40%	
	distance-to-(0,1)	0.4194	0.5106	0.5424	0.5547	0.5909	
	sensitivity	74.40%	51.30%	48.00%	44.50%	42.10%	DB2
	specificity	69.57%	87.63%	89.97%	91.58%	92.68%	
	distance-to-(0,1)	0.3976	0.5025	0.5296	0.5613	0.5836	
	sensitivity	79.70%	52.90%	51.00%	48.00%	44.90%	DB4+DB2
specificity	72.49%	89.59%	92.03%	93.49%	94.36%		
distance-to-(0,1)	0.3419	0.4824	0.4964	0.5241	0.5539		
	sensitivity	66.25%	49.38%	46.75%	45.25%	42.50%	DB4
	specificity	67.87%	87.11%	90.01%	91.71%	92.65%	

Condition17	distance-to-(0,1)	0.4660	0.5224	0.5418	0.5537	0.5797	DB2
	sensitivity	72.42%	48.83%	45.50%	41.83%	38.08%	
	specificity	70.22%	87.79%	90.27%	92.02%	93.31%	
	distance-to-(0,1)	0.4059	0.5260	0.5536	0.5871	0.6228	DB4+DB2
	sensitivity	76.80%	51.20%	48.30%	44.20%	42.90%	
	specificity	72.97%	89.85%	92.52%	94.01%	94.83%	
	distance-to-(0,1)	0.3562	0.4984	0.5224	0.5612	0.5733	

Table 3.20 reveals a lot of information. Huge disparity of performance exists between some cases. Sensitivities and specificities vary from 30% to 95%. There is an obvious trend affected by the choice of number ‘k’ of the nearest neighbor: when ‘k’ increases, the sensitivity drops while the specificity rises. There is no clear evidence how ‘k’ affects the measurement distance-to-(0,1). Yet we can still observe that under fourteen conditions, the smallest distance-to-(0,1) values occur when ‘k’ equals to one; under two conditions (‘conditon1 & 9’), the smallest distance-to-(0,1) values occur when ‘k’ equals to three; under one condition (‘conditon8’), the smallest distance-to-(0,1) occurs when ‘k’ equals to nine. The tests also confirmed that employing the dual-wavelet features can still benefit the classification, even in fuzzy cases, except in ‘conditon5, 8 & 14’, where using features from DB2 yields the best performance.

If distance-to-(0,1) is used as the criterion to evaluate the performance, the best result overall is yielded by Condition16 with dual-wavelet feature plus 1-nearest-neighbor choice. The distance-to-(0,1) of Condition16 is 0.3419, while the the sensitivity reaches 79.7% and the specificity is 72.49%. Condition4, 6, 7, 14 & 15 with the same choice also yield decent results.

Table 3.21 compares the top fuzzy results with the crisp results listed in Section 3.2.4.1.1. When the test is performed on all data, the results yielded by Condition1 and Condition2 are similar to the crisp result. With appropriate initialization of the membership function, the sensitivity of the fuzzy test Condition16 is 2.67% higher than the benchmark crisp result; the specificity is 2.29% higher. Condition4 and Condition15 also show certain degree of improvement. When the quality of the test data is improved by removing controversial data, Condition4 yielded the best result of 79.57% sensitivity and 75.80% specificity, which is respectively 0.67% and 2.34% higher than those

in the benchmark crisp test. The increase of sensitivity is not high when the “controversial” data are removed .

Table 3.21: Comparison between selected crisp and fuzzy results

		based on all data	discard controversial data (retained vote: 0, 4, 5, 6)
benchmark crisp case (k=3)	sensitivity	77.03%	78.90%
	specificity	70.20%	73.46%
	dist.	0.3763	0.3390
Condition2 (k=1)	sensitivity	74.13%	75.88%
	specificity	71.94%	74.07%
	dist.	0.3817	0.3542
Condition1 (k=3)	sensitivity	69.25%	69.36%
	specificity	77.90%	81.91%
	dist.	0.3787	0.3558
Condition16 (k=1)	sensitivity	79.70%	79.00%
	specificity	72.49%	74.29%
	dist.	0.3419	0.3319
Condition15 (k=1)	sensitivity	76.93%	79.25%
	specificity	73.71%	75.62%
	dist.	0.3498	0.3201
Condition4 (k=1)	sensitivity	76.44%	79.57%
	specificity	73.75%	75.80%
	dist.	0.3527	0.3167

3.2.4.2 Fuzzy c-Means

3.2.4.2.1 Clustering Results of Fuzzy c-means

Using fuzzy c-means, a clustering method, some of the membership function initialization strategies proposed in Section 3.2.3 yield identical clusters when performing on annotation set ‘phase2’, as shown in Table 3.22 and 3.23.

Clearly, the distribution of the data does not show any correlation between the clusters and the paroxysmal types. Moreover, the distribution indicates that there is a main cluster that includes most of the data, with sparse data in another small clusters.

Table 3.24 shows an additional test on the 200-patient set.

Table 3.22: Number of data vectors related with the c th mean in the 2-mean case on ‘phase2’

	DB4		DB2		DB4+DB2	
	Mean1	Mean2	Mean1	Mean2	Mean1	Mean2
AEP:6	0	20	20	0	20	0
AEP:5	0	17	17	0	17	0
AEP:4	0	47	47	0	47	0
AEP:3	0	32	32	0	32	0
AEP:2	0	53	53	0	53	0
AEP:1	0	125	125	0	125	0
AP	15	796	790	21	788	23
NEP	0	757	757	0	757	0

Table 3.23: Number of data vectors related with the c th mean in the 3-mean case on ‘phase2’

	DB4			DB2			DB4+DB2		
	Mean1	Mean2	Mean3	Mean1	Mean2	Mean3	Mean1	Mean2	Mean3
AEP:6	2	0	18	1	0	19	1	0	19
AEP:5	1	0	16	0	0	17	0	0	17
AEP:4	3	0	44	0	0	47	0	0	47
AEP:3	4	0	28	0	0	32	0	0	32
AEP:2	4	0	49	0	0	53	0	0	53
AEP:1	6	0	119	0	0	125	0	0	125
AP	56	10	745	30	5	776	33	4	774
NEP	26	0	731	3	0	754	3	0	754

Table 3.24: Number of data vectors related with the c th mean in the 2-mean case on 200-p set

	DB4		DB2		DB4+DB2	
	Mean1	Mean2	Mean1	Mean2	Mean1	Mean2
AEP:6	0	36	0	36	0	36
AEP:5	1	41	2	40	2	40
AEP:4	0	64	0	64	0	64
AEP:3	0	55	0	55	0	55
AEP:2	0	93	0	93	0	93
AEP:1	1	260	2	259	2	259
AP	17	1129	25	1121	22	1124
NEP	7	1326	6	1327	7	1326

Table 3.25: Number of data vectors related with the c th mean in the crisp 2-mean case on ‘phase2’

	DB4		DB2		DB4+DB2	
	Mean1	Mean2	Mean1	Mean2	Mean1	Mean2
AEP:6	20	0	0	20	0	20
AEP:5	17	0	0	17	0	17
AEP:4	47	0	0	47	0	47
AEP:3	32	0	0	32	0	32
AEP:2	53	0	0	53	0	53
AEP:1	125	0	0	125	0	125
AP	797	14	21	790	21	790
NEP	757	0	0	757	0	757

Table 3.26: Number of data vectors related with the c th mean in the crisp 3-mean case on ‘phase2’

	DB4			DB2			DB4+DB2		
	Mean1	Mean2	Mean3	Mean1	Mean2	Mean3	Mean1	Mean2	Mean3
AEP:6	0	1	19	0	1	19	0	1	19
AEP:5	0	1	16	0	0	17	0	0	17
AEP:4	0	1	46	0	0	47	0	0	47
AEP:3	0	1	31	0	0	32	0	0	32
AEP:2	0	1	52	0	0	53	0	0	53
AEP:1	0	3	122	0	0	125	0	0	125
AP	10	38	763	5	29	777	4	28	779
NEP	0	8	749	0	2	755	0	2	755

Table 3.27: Number of data vectors related with the c th mean in the 2-mean case on 200-p set

	DB4		DB2		DB4+DB2	
	Mean1	Mean2	Mean1	Mean2	Mean1	Mean2
AEP:6	36	0	36	0	36	0
AEP:5	41	1	40	2	41	1
AEP:4	64	0	64	0	64	0
AEP:3	55	0	55	0	55	0
AEP:2	93	0	93	0	93	0
AEP:1	260	1	259	2	259	2
AP	1129	17	1124	22	1125	21
NEP	1326	7	1327	6	1327	6

3.2.4.2.2 Comparison to Crisp c-Means

To confirm the fuzzy clustering result, the crisp c-means is applied. Table 3.25 and Table 3.26 show the results of crisp c-means.

The results of 2-means clustering of crisp cases and fuzzy cases are exactly the same. There are slightly differences between the results of 3-means clustering. However, this does not undermine the fact that with the current features, there is no cluster that can represent a certain paroxysmal type.

Table 3.27 is the result of an additional test on the 200-patient set, where the same trend of clustering behavior as the fuzzy c-means is observed.

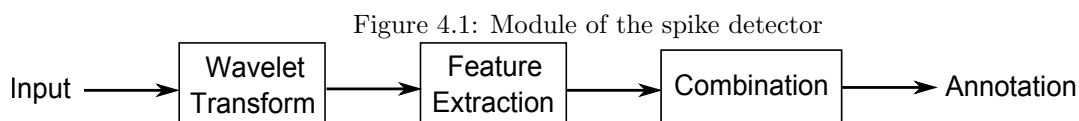
Chapter 4

Yellow-Box Detection

4.1 Methodology for Design of Detection of Yellow-Box

Section 2.2 explains the acquisition procedures of the data for training and testing. One of the most time-consuming procedures in obtaining YBs is visual inspection by experts. It is desirable to develop a reliable automatic YB detector to increase the efficiency of the research. In this chapter, we intend to imitate the entire YB production procedure by human experts. This YB detector is essentially a preliminary spike detector. It aims to detect candidate paroxysmal events and generate corresponding annotations for following ETs classification.

The YB detector includes five modules as shown in Figure 4.1.



The input data in the first module are 30-second EEG segments. The segments are derived from the 100-patient dataset in Section 2.2. The following two montages are adopted:

1. 21-channel Referential Average;
2. 18-channel-pair Bipolar AP Typical.

There are 2100 30-second segments if the data of referential average montage is used for test and evaluation, and there are 1800 30-second segments if the data of bipolar AP typical montage is used

for test and evaluation.

In our previous study we found that the Daubechies wavelet of order 4 (DB4) and order 2 (DB2) mother wavelets yielded features with better performance for paroxysmal classification than other commonly used mother wavelets (DB 5, DB 20, bior1.3, bior 1.5) in ET detection research [68]. Indiradevi's study suggested that the DB4 is particularly useful for ET detection research since it obtains the highest correlation coefficients with the epileptic spike signal among the available wavelet bases in the Matlab toolbox [32]. Guler suggested that for a particular application, tests should be performed in advance with different types of wavelets, and the one which gives maximum efficiency should be selected. In his opinion, the smoothing feature of DB2 is more suitable to detect changes of the EEG signals [25]. In this study, we decided to employ only DB2 and DB4 mother wavelets to perform WT in the second module of the YB detector.

4.1.1 Plain Detection

After wavelet decomposition, proper features need to be extracted from raw wavelet coefficients to reduce the computational complexity. A variety of features related to epilepsy have been suggested by previous studies. Three mostly used types of feature are

1. Raw/normalized wavelet coefficients;
2. Square of the wavelet coefficients; and
3. Entropy.

In our study, we found that most of the entropy features are used in epilepsy seizure detection. Epilepsy seizures are long-lasting signals and can be viewed as stationary at some point. Epilepsy seizures are not suitable for transient signal like ET. We employed entropy features in our preliminary studies yet they yielded low performance. The square of the wavelet coefficient is used since it only concerns the spikiness of the signal. It fits the description of the ET components scattering in different decomposition subbands.

The detection result yielded by a single feature is less reliable. Combination of individual (subband/bipolar) decisions is necessary in order to achieve better results. There are two basic types of combination:

1. Linear combination;

2. Nonlinear combination.

The commonly used strategies in combination are listed as follows [38]:

1. Product rule;
2. Sum rule;
3. Max rule;
4. Min rule;
5. Medium rule; and
6. Voting, this is a flexible strategy. Three forms of voting are implemented in Section 4.1.1.1:
 - (a) AND decision by detail subband $D4$ and $D5$ (Indiradevi's suggestion) [31];
 - (b) Majority vote decision by all detail subbands (votes ≥ 3 out of 5 votes); and
 - (c) Weighted vote decision by all detail subbands (votes \geq a preselected threshold). To accomplish the combination of weighted decisions, besides conventional weights (e.g., equal weights), optimization of weights is required. There are several methods of weights determination and optimization:
 - Use the standard deviations of the errors [38];
 - Density-based weighting (require knowledge of prior probability) [59];
 - Unified approach: correlated errors and general coupling [59]; and
 - Belief integration using belief value [9].

Belief integration suggested by Chen is a practical and straightforward method. Further details are discussed in Section 4.1.1.2.

4.1.1.1 Synopsis of Indiradevi's Algorithm (2007)

Indiradevi suggested a spike detection scheme using wavelet coefficients in the long-term EEG recording [31]. In this scheme, the individual signals (sampling rate 256Hz) are decomposed into k ($k = 6$ in Indiradevi's case) scales using a proper mother wavelet (DB4 in Indiradevi's case). In our study, the sampling frequency of the EEG signals is 256 Hz. The highest frequency

component that the signal could contain, according to Nyquist theorem, would be 128 Hz. A five-level decomposition will satisfy the requirement that the dominant frequency components of the signal are retained in the wavelet coefficients. The corresponding frequency ranges of the subbands with a 256Hz sampling rate are listed in Table 4.1:

Table 4.1: Corresponding frequency range of each subband in detection

Subband	Frequency Range
D1	64Hz ~ 128Hz
D2	32Hz ~ 64Hz
D3	16Hz ~ 32Hz
D4	8Hz ~ 16Hz
D5	4Hz ~ 8Hz
A5	0Hz ~ 4Hz

The optimal resolution to analyze the epileptiform activities corresponds to the frequency band 4 to 32 Hz [31]. To minimize contamination by non-epileptiform high frequency signals like muscle artifacts, Indiradevi focused on sub-bands 4 and 5 (4-8Hz & 8-16Hz).

Indiradevi did not suggest any sophisticated features. Instead they used the square of wavelet coefficients in subband $D4$ and $D5$ ($d_{j,k}^2$, $j = 4, 5$). If the square of wavelet coefficient at one time is above a pre-determined threshold level, this point is marked as a spike [31].

The pre-determined threshold is computed as:

$$T_j = C \times std(D_j)S_j \quad (4.1)$$

where

- $S_j = 2^j / \Delta\psi_j$;
- $\Delta\psi_j = \max\psi_{j,k}(t) - \min\psi_{j,k}(t)$, $\psi(t)$ is the wavelet function;
- D_j is the reconstructed detail coefficients; and
- Constant C is derived from the average value of standard deviations of the whole dataset.

In fact the components of ET are distributed in all subbands and the contribution of wavelet coefficients in different subbands needs to be quantified and proper weights need to be assigned to each subband, respectively. Section 4.1.1.2 introduces a belief value based optimization method.

4.1.1.2 Subband Weight Optimization by Belief Value

A confusion matrix is defined as [9]:

$$CM_k = \begin{pmatrix} n_{11}^k & n_{12}^k & \dots & n_{1m}^k \\ n_{21}^k & n_{22}^k & \dots & n_{2m}^k \\ \dots & \dots & \dots & \dots \\ n_{m1}^k & n_{m2}^k & \dots & n_{mm}^k \end{pmatrix}$$

The element n_{ij}^k means that n_{ij}^k samples of class i are classified as class j by the k th classifier. The number of samples in class i is: $n_i^k = \sum_{j=1}^m n_{ij}^k$.

In our case, notice that the number n_{ij}^k can be either number of annotations or number of samples in total annotations. Belief value is defined as [9]:

$$b_k^{ij} = b_k(x \in \text{class } i \mid \text{classification decision is class } j) = \frac{n_{ij}^k/n_i^k}{\sum_{t=1}^m n_{tj}^k/n_t^k}. \quad (4.2)$$

In our case, we use Indiradevi's method to detect spikes with features from single subband. Each subband can be viewed as one detector (classifier). Then we can compute the belief value b_k of the k th detector ($k = D1, D2, D3, D4, D5$). The belief value which reflects the capability of spike detection is b_k^{11} . Normalize the vector $[b_{D1}^{11} \ b_{D2}^{11} \ b_{D3}^{11} \ b_{D4}^{11} \ b_{D5}^{11}]^T$. This vector will be an optimal weight combination.

If we maximize detected number of annotations, the weight combination is:

$$\mathbf{db4} = [0.2062, 0.3186, 0.2493, 0.1501, 0.0759]^T$$

$$\mathbf{db2} = [0.2568, 0.3010, 0.2281, 0.1365, 0.0776]^T$$

$$\mathbf{db4+db2} = [0.1936, 0.2991, 0.2340, 0.1409, 0.0712, 0.2725, 0.3194, 0.2420, 0.1449, 0.0823]^T/2.$$

4.1.1.3 Methodology of Performance Evaluation of Plain Detection

Three types of parameters are used to evaluate the performance of the YB detector:

1. The number of true positive (TP), false negative (FN), false positive (FP), sensitivity ($\#TP/\#\text{all-marked-paroxysmal-event}$) and selectivity ($\#TP/(\#TP+\#FP)$) for 2565 positive paroxysmal

annotations (83 ETs and 2482 non-ETs); specificity ($\#TN/\#all\text{-negative}$) for 2998 negative annotations;

2. Length of overlap/non-overlap of an automatic detected annotation with the nearest expert-marked annotation; and
3. Cost of transform between a detected annotation and a corresponding expert-marked annotation. In this case we use the sum of distances of expert-marked-annotation-start-point to detected-annotation-start-point and expert-marked-annotation-end-point to detected-annotation-end-point; If a detected annotation did not have overlap with any expert-marked annotation, the cost is the distance between the detected-annotation-start-point and the detected-annotation-end-point to make this annotation disappear, which equals to the length of this annotation.

4.1.2 Implementation of Artificial Neural Network with a Pruning Procedure

Indiradevi's algorithm is based on a pre-determined threshold, which is a simple straightforward decision rule. To pursue possible improvement in performance, a more complex decision boundary is developed.

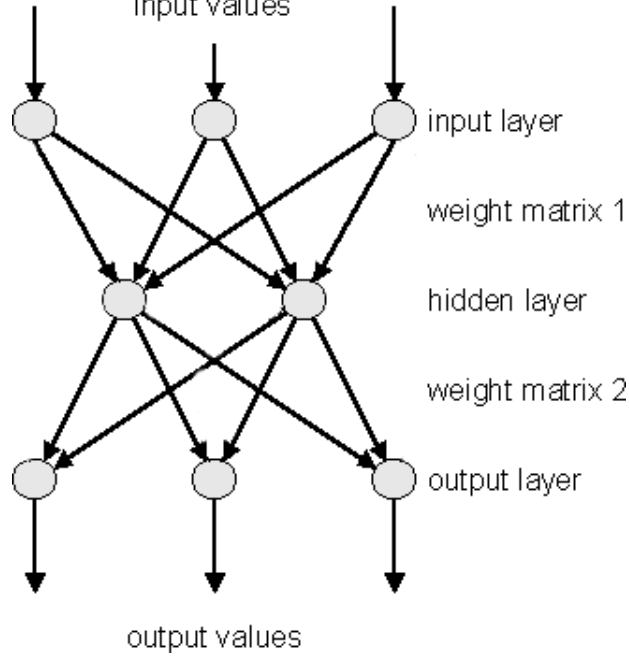
Artificial Neural Networks are known to facilitate better design and implementation of machine learning. ANNs can learn from experience and implement complex decision surfaces, although the training period can be time consuming [54]. The multilayer feedforward (MLFF) network is one of the most commonly used ANN structures and is adopted to accomplish our detection task. The Backpropagation (BP) weights-updating algorithm is used to update the status of the network. Figure 4.2 ¹ illustrates a simple MLFF net using BP algorithm.

The output of each unit is a linear or non-linear function $f(x)$ of the units in the previous layer. To avoid shortcomings yielded by a linear output function, a non-linear differentiable activation function (logarithm sigmoid, tangent sigmoid, etc) is used:

$$o_j^p = f(net_j) \tag{4.3}$$

¹From <http://www.geoneurale.com/MultilayerPerceptrons.htm>

Figure 4.2: Multilayer feedforward network using back-propagation algorithm



where

$$net_j = \sum_i w_{ji} o_i^p \quad (4.4)$$

is the weighted linear combination of the outputs of all the units in previous layer.

The BP algorithm uses gradient descent to derive weights that minimizes the output error

$$E^p = \frac{1}{2} \sum_j (t_j^p - o_j^p)^2. \quad (4.5)$$

For a case training by epoch, the formulation is

$$E = \sum_p E^p. \quad (4.6)$$

When updating the weights, the algorithm starts from the output layer and then traces to the hidden layer. Calculate the partial derivative of the output error with respect to weight w_{ji} (weight between output unit j and hidden layer unit i)

$$\frac{\partial E^p}{\partial w_{ji}} = \frac{\partial E^p}{\partial o_j^p} \frac{\partial o_j^p}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}. \quad (4.7)$$

where

$$\frac{\partial E^p}{\partial o_j^p} = -(t_j^p - o_j^p) \quad (4.8)$$

$$\frac{\partial o_j^p}{\partial net_j} = f'(net_j) \quad (4.9)$$

$$\frac{\partial net_j}{\partial w_{ji}} = o_i^p. \quad (4.10)$$

According to general delta rule, the weight correction is

$$\Delta^p w_{ji} = \epsilon \delta_j^p o_i^p \quad (4.11)$$

where

$$\delta_j^p = -\frac{\partial E^p}{\partial o_j^p} \frac{\partial o_j^p}{\partial net_j} \quad (4.12)$$

and ϵ is the learning rate.

When the unit j is in the hidden layer, Equation 4.8 becomes

$$\begin{aligned} \frac{\partial E^p}{\partial o_j^p} &= \sum_k \frac{\partial E^p}{\partial net_k} \frac{\partial net_k}{\partial o_j^p} \\ &= -\sum_k \delta_k^p w_{kj} \end{aligned} \quad (4.13)$$

where k is the index of units in the output layer. The correction then is

$$\Delta^p w_{ji} = \epsilon \delta_j^p o_i^p \quad (4.14)$$

where

$$\delta_j^p = (\sum_k \delta_k^p w_{kj}) f'(net_j). \quad (4.15)$$

A proper learning rate is essential. Result divergence can be caused by a too high learning rate or undertraining caused by a too small learning rate, especially in our case where the boundaries between different types of EEG signal can be extremely complex in vector space [27]. However, there is no single, evident learning rate choice. Instead of fixing the learning rate in the entire training

process, we implement an adaptive learning rate that guarantees convergence theoretically [6]:

$$\eta_a = \mu \frac{\|\tilde{y}\|^2}{\|\mathbf{J}_P^T \tilde{y}\|^2} \quad (4.16)$$

where $\tilde{y} = y_d^p - y^p$ and $\mathbf{J}_P^T = (\partial y^p / \partial \mathbf{W})$ (y_d is the desired output; y is the actual output; \mathbf{W} is the weight vector) [6].

4.1.2.1 Calculation of S_{ij} , the Sensitivity of the Error Function

When the wavelet decomposition coefficients are selected as features for implementation of ANN, the issue of high-dimensional input vector emerges. The desired DWT decomposition level for YBs detection varies from four to six, where five to seven subbands will be produced. Researchers are seeking more than one feature in each subband to characterize ETs [25][32]. The incorporation of multiple features from multiple subbands eventually produces a high dimensional input vector, which is computationally expensive. On the other hand, the contributions of the subbands to YB detection are nonequivalent. They are affected by the relevance of the corresponding frequency bands. In sequence, the most important frequency bands are: (1) 14.3-50 Hz, related with spikes; (2) 5-14.3 Hz, related with sharp waves; and (3) 2.8-6.7 Hz, related with slow wave. The type of features also contributes to the performance in varying degrees. To reduce the computational complexity, evaluation of the influence of various candidate subbands and feature types is necessary.

Instead of brute-force determination of the performance of individual features and/or subbands, we conduct small-scale experiments (instead of testing on the real-time recording data, test only on the existing YBs while each of them yielding only one data vector) via a feedforward neural network and train it with features of Set #1 in Section 3.1.1.2. A strategy typically used for network unit pruning is applied here: Estimate the sensitivity of the network mapping error function to each network weight (S_{ij}) associated with the input layer. By focusing on the weights of the input units, the contributions of the corresponding input features are indirectly determined .

The idea of estimating the sensitivity of the ANN mapping error function (during training) to weight elimination was proposed by Mozer and Smolensky [44] as a weight-centric network pruning procedure. The sensitivity of the mapping error with respect to any network weight w_{ij} is defined as

$$S_{ij} = E(w_{ij} = 0) - E(w_{ij} = w_{ij}^f) \quad (4.17)$$

where the mapping error is defined as $E = \sum_p \sum_k (o_k^p - t_k^p)^2$ for output k and training set pattern p and w_{ij}^f is the final value yielded by training process. Karnin [35] reformulated the S_{ij} in Equation 4.17 as:

$$S_{ij} = -\frac{E(w_{ij}^f) - E(0)}{w_{ij}^f - 0} w_{ij}^f. \quad (4.18)$$

Considering that a typical learning process starts with each weight initialized to some random small values rather than zero to avoid premature saturation (also because initialization of all weights to zero is a suboptimal training strategy [54]), Equation 4.17 can be approximated by the initial state

$$S_{ij} \approx -\frac{E(w_{ij}^f) - E(w_{ij}^i)}{w_{ij}^f - w_{ij}^i} w_{ij}^f. \quad (4.19)$$

For a network with d weights $u_1 \dots u_{d-1}$ and w_{ij} , consider only w_{ij} changes from the zero state (A) to the final state (F) while the other weights remain in their final states. The value of the error function will decrease along the gradient from $w_{ij} = 0$ to $w_{ij} = w_{ij}^f$ as

$$E(w_{ij} = w_{ij}^f) - E(w_{ij} = 0) = \int_A^F \frac{\partial E(u_1^f, \dots, w_{ij})}{\partial w_{ij}} dw_{ij}. \quad (4.20)$$

Using the initial state approximation (I instead of A), Equation 4.20 yields:

$$E(w_{ij} = w_{ij}^f) - E(w_{ij} = 0) \approx \int_I^F \frac{\partial E(u_1^f, \dots, w_{ij})}{\partial w_{ij}} dw_{ij}. \quad (4.21)$$

In practice, the integral operation in Equation 4.21 can be approximated by the summation of the correction of weights in each epoch. Substituting the numerator of Equation 4.19 with the approximation of Equation 4.21 yields:

$$\tilde{S}_{ij} = -\sum_0^{N-1} \frac{\partial E}{\partial w_{ij}}(n) \Delta w_{ij}(n) \frac{w_{ij}^f}{w_{ij}^f - w_{ij}^i} \quad (4.22)$$

where N is the number of training epoch and Δw_{ij} is the weight correction in each step.

In a network trained with backpropagation algorithm, $\partial E / \partial w_{ij}$ in each step can be obtained

directly using the generalized delta rule [54]:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= -\delta_i^p o_j^p \\ &= \begin{cases} -(t_i^p - o_i^p) f'_j(\text{net}_i^p) o_j^p, & \text{outputlayer} \\ -(\sum_n (-\delta_n^p w_{ni})) f'_j(\text{net}_i^p) o_j^p, & \text{hiddenlayer.} \end{cases} \end{aligned} \quad (4.23)$$

If a weight has a relatively small estimated S_{ij} , the mapping error of the net will not decrease significantly if this weight is removed. Furthermore, if all the weights linked to one node (unit) have small S_{ij} , this node is less likely to contribute to the classification performance of the neural net and thus can be pruned. The significance is that if a to-be-pruned node is an input node, it indicates that the feature related to this input node is less essential.

4.1.2.2 Dingle's Feature Choice: Morphology and Background

Dingle [15] believed that the only way to separate epileptiform from nonepileptiform waves is to make use of a wide spatial and temporal context. He then proposed a set of features capable of detecting a high proportion of epileptiform transients and providing context information. All these features are based on morphological traits of target waves or background waves. Acir [1] and Liu [41] also suggested other morphology based features.

Collecting up their proposals, the characteristics of a wave is usually depicted by the following parameters:

1. Duration:
 - (a) first half wave (apex to apex) duration (FHWD) [1];
 - (b) second half wave (apex to apex) duration (SHWD) [1];
 - (c) wave duration (sum of two half wave durations) [41]; and
 - (d) duration between turning points (turning point is defined as the point where the slope has the maximum amplitude on the half wave) [15] [41]
2. Amplitude:
 - (a) first half wave amplitude (FHWA) [1];

- (b) second half wave amplitude (SHWA) [1];
- (c) amplitude from peak to the first turning point [41];
- (d) amplitude from peak to the second turning point [41]; and
- (e) amplitude over a floating mean (the average EEG value over 75ms centered on the peak) [15]

3. Slope:

- (a) first half wave slope (FHWS) [1];
- (b) second half wave slope (SHWS) [1];
- (c) slope at point k [41]

$$sl(k) = (w(k+1) - w(k-1))/2; \text{ and}$$

- (d) consecutive slope at point k [15]

$$sl(k) = w(k) - w(k-1);$$

4. Sharpness:

- (a) sharpness at point k (changing rate of the slope at the peak point) [41]

$$sh(k) = (sl(k+1) - sl(k-1))/2;$$

- (b) sharpness of the spike [15]

$$SH = SHWS - FHWS.$$

In total, we selected ten features based on the above proposals and the knowledge of EEG data, including five spike-related features and five background-related features.

- The following five features are selected to depict the spikes:
 1. Wave duration;
 2. Duration between turning points;
 3. Amplitude over a floating mean;

4. Sharpness at point k ; and
5. Sharpness of the spike
- Theoretically, ETs are defined to be clearly distinguished from background activity. The following five measures of the background activity are calculated as a compensation [15]:
 6. Background amplitude (the average difference between the EEG and the floating mean);
 7. Background slope (the average magnitude of the consecutive slope);
 8. Background duration (the average peak-to-peak duration of the halfwaves); and
 9. Background rhythmicity, defined by two parameters:
 - (9a) the coefficient of variation (standard deviation/mean) of halfwave durations;
 - (9b) the coefficient of variation of halfwave amplitudes

4.1.3 Clustering of Yellow-Boxes

4.1.3.1 Grouping

According to experts' opinion, when an event occurs during a temporal interval, its activity can emerge in several channels on one or even more montages, which will yield multiple YB candidates. Under this circumstance, only one candidate with the maximum amplitude will be selected through experts' visual inspection on the amplitude of the candidates appeared in relevant channels.

In a real-time simulation, the machine also has to screen redundant candidates produced by the detector, as shown in Figure 1.2. The function is fulfilled by the following algorithm:

Algorithm 1:

BEGIN

Input all the YB candidates from all channels, $\{Y_i\}$, $i = 1, 2, \dots, m$.

Initialize group#1.

Let Y_1 be a member of group#1.

Initialize $n = 1$, $i = 2$, $k = 1$.

Initialize desired overlap percentage as $share = 50\%$.

DO UNTIL ($i > m$)

Set the status of Y_i as 'free'.

```

DO UNTIL (  $Y_i$  query all the  $n$  current existing groups )
  Compute the temporal overlap of  $Y_i$  with each member  $X_{kj}$ ,  $j = 1, 2, \dots, p$  in group# $k$ .
  IF ( the temporal overlap of  $Y_i$  and every  $X_{kj}$  is over 'share' of both the length of  $Y_i$  and
 $X_{kj}$  ) THEN
    Let  $Y_i$  be a member of group# $k$ .
    Set the status of  $Y_i$  as 'member'.
  END IF
END DO UNTIL
IF ( the status of  $Y_i$  stays at 'free' ) THEN
  Set  $n = n + 1$ .
  Build group# $n$ .
  Let  $Y_i$  be a member of group# $n$ .
END IF
Set  $i = i + 1$ .
END DO UNTIL
END

```

Through the implementation of *Algorithm 1*, annotation candidates are grouped into clusters, in which every two annotations share at least 50% (this percentage can be adjusted as needed) of their temporal intervals. Notice that one annotation can join several groups depending on the complexity of the distribution of candidates.

After grouping, the machine rules out redundant annotations by examining their amplitude. The following algorithm is implemented to realize this process:

Algorithm 2:

```

BEGIN
  Input all the groups: {group# $k$ },  $k = 1, 2, \dots, p$ .
  Initialize  $k = 1$ .
  DO UNTIL (  $k > p$  )
    Determine the maximum temporal interval  $len_k$  that all members,  $X_{kj}$ ,  $j = 1, 2, \dots, l$ , share in
    group# $k$ .
  
```



```

Compute the energy  $E_j = \|X_{kj}(len_k)\|^2, j = 1, 2, \dots, l$ .
Determine  $q$  that  $E_q = \max E_j, j = 1, 2, \dots, l$ .
Set the status of  $X_{kq}$  as 'Yellow Box'.
Set  $k = k + 1$ .
END DO UNTIL
END

```

By implementing *Algorithm 2*, the candidate whose amplitude energy is the highest in the mutually overlapped temporal interval within the group, is selected as YB.

4.1.3.2 Merging and Discarding

When annotating YBs, human experts are able to see and understand the context, and thus they tend to mark related events. Machine tends to find temporal points at which typical features are yielded. These temporal points are not necessary to be consecutive. It is also possible that not all the traits of a event fit the ET standard. One event is likely to be annotated as several YBs. Therefore, YBs belonging to the same event are needed to be merged while very short outliers should be screened. A practical way is to set two thresholds, one for merging close YBs and the other for discarding short outliers. Notice three values given by Section 1.1 are important: (1) 20ms, the inferior limit of lasting time of ET; (2) 70ms, the superior limit of lasting time of ET and also the inferior limit of lasting time of slow wave; and (3) 200ms, the superior limit of lasting time of slow wave). With 256Hz sampling rate, these three values respectively correspond to: (1) 5 temporal points; (2) 18 temporal points; and (3) 52 temporal points. We adopt the inferior limit of lasting time of ET, 5 points, as the threshold of discarding outliers. Both choices 18 and 52 are tested as the threshold of merging in the final simulation.

4.1.4 Full-scale Real-time Simulation

In the real world, detector's performance much be generalizable. The ultimate purpose of Chapter 4 is to complete simulations on consecutive EEG data with mutually independent training and test populations, as it is in the real world.

The simulation is designed as follows:

1. Split the 100-patient dataset by patient, using 10-fold cross-validation strategy; due to the unbalanced distribution of AEP annotations (AEPs only exist in 31 patients while nonAEPs exist in all 100 patients), a restriction that every fold must include at least one patient who can provide AEP annotations is applied;
2. Choose desired features and ratio of AEP:nonAEP:negative data;
3. Initialize ANN and choose training parameters;
4. Train the ANN for every fold;
5. In each fold, input the EEG of test population; Record the outcome YB candidates;
6. Choose merging and discarding threshold for the candidates;
7. Choose the overlap rate to group;
8. Yield the final YBs.

The simulation will be measured by sensitivity, specificity and selectivity. The definitions of sensitivity and specificity are given in Section 3.1.1.5.4. The implication of TP, FP, TN, FN and selectivity is defined as:

TP: expert marked YB that has effective overlap with any machine created YBs; a effective overlap happens when two YBs share a certain temporal interval and they are on the same channel at the same time;

FP: machine created YB that does not have any effective overlap with either expert marked YB;

TN: expert marked negative annotation that does not have any effective overlap with either machine created YB;

FN: expert marked negative annotation that has effective overlap with any machine created YBs;

Selectivity = $TP / (TP + FP)$;

4.2 Results and Evaluation of Yellow-Box Detection

4.2.1 Implementation of Indiradevi's Algorithm

In this section, 16 parameters are used to evaluate performance of the YB detector. Each parameter measures a specific aspect, listed as below:

Eva#1 average length of expert-marked annotations overlapped with machine-detected YBs;

Eva#2 average length of expert-marked annotations missed by detector;

Eva#3 average length of total expert-marked annotations;

Eva#4 average length of overlaps between TP and expert-marked annotations;

Eva#5 average length of overlaps between all detected annotations and expert-marked annotations;

Eva#6 average of the rate of overlap-length/corresponding-expert-marked-annotation-length of the TP among the machine-detected YBs;

Eva#7 average length of non-overlaps of the TP among the machine-detected YBs;

Eva#8 average length of the FP, which refers to the machine-detected YBs having no overlap with any expert-marked annotations;

Eva#9 average length of the FN, which refers to the expert-marked annotations having no overlap with any machine-detected YBs;

Eva#10 average length of non-overlap of all the machine-detected YBs;

Eva#11 average of the rate of non-overlap-length/expert-marked-annotation-length of the TP among the machine-detected YBs;

Eva#12 average cost of transform of the TP among the machine-detected YBs;

Eva#13 average cost of transform of the FP among the machine-detected YBs;

Eva#14 average cost of transform of the FN among the expert-marked annotations;

Eva#15 average cost of transform of all the annotations;

Eva#16 average of the rate of cost-of-transform/corresponding-expert-marked-annotation-length of the TP among the machine-detected YBs.

Respectively, Table 4.2, Table 4.3, Table 4.4 and Table 4.5 list specific results of YB detector based on Indiradevi's algorithm (shown in Section 4.1.1.1) with four weight choices. In Indiradevi's algorithm, the wavelet coefficients are transformed into binary signal after applying the threshold computed by Equation 4.1. The entries of $WTcoeff$ below are either 0 (wavelet coefficient is smaller than the threshold) or 1 (wavelet coefficient is larger than the threshold)

$$WTcoeff = [D1 \ D2 \ D3 \ D4 \ D5 \ A5]. \quad (4.24)$$

For a 5-level wavelet decomposition using DB2 or DB4, the weight vector is:

$$weight_{DB4} = [WD1_{DB4} \ WD2_{DB4} \ WD3_{DB4} \ WD4_{DB4} \ WD5_{DB4} \ WA5_{DB4}] \quad (4.25)$$

$$weight_{DB2} = [WD1_{DB2} \ WD2_{DB2} \ WD3_{DB2} \ WD4_{DB2} \ WD5_{DB2} \ WA5_{DB2}]. \quad (4.26)$$

Decision is made by

$$WTcoeff \odot weight \geq Th. \quad (4.27)$$

The weights of the entries in the vector are assigned as follows:

- equal weights for all subbands except approximation subband:

$$weight_{DB4} = [0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2 \ 0];$$

$$weight_{DB2} = [0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2 \ 0];$$

$$Th = 0.201;$$

- AND decision of subband $D4$ and $D5$:

$$weight_{DB4} = [0 \ 0 \ 0 \ 0.2 \ 0.2 \ 0];$$

$$weight_{DB2} = [0 \ 0 \ 0 \ 0.2 \ 0.2 \ 0];$$

$$Th = 0.201;$$

- optimal weight #1:

$$weight_{DB4} = [0.1491 \ 0.2824 \ 0.2321 \ 0.1952 \ 0.1412 \ 0];$$

$$weight_{DB2} = [0.1660 \quad 0.2570 \quad 0.2242 \quad 0.1875 \quad 0.1653 \quad 0];$$

$$Th = 0.201;$$

- optimal weight #2:

$$weight_{DB4} = [0.2062 \quad 0.3186 \quad 0.2493 \quad 0.1501 \quad 0.0759 \quad 0];$$

$$weight_{DB2} = [0.2568 \quad 0.3010 \quad 0.2281 \quad 0.1365 \quad 0.0776 \quad 0];$$

$$Th = 0.201.$$

The weight of $A5$ is set to zero since the principal components of $A5$ are DC and low frequency noise.

We noticed that in Table 4.2, Table 4.3, Table 4.4 and Table 4.5, none of the sensitivities is above 80%. By separating the ETs' sensitivity from the non-ETs', we observed that the sensitivity values of ETs are 30% higher than those of the overall performance, while the sensitivity values of non-ETs are slightly (5% or lower) below those of the overall performance. The $D4-D5$ -AND weight yields the lowest result. The two optimized weights yield relatively good sensitivity, while the equal weight and $D4-D5$ -AND weight yield 6% higher selectivity than the optimized weights do. The average overlap of $D4-D5$ -AND weight is 10 samples less than the other three cases and its non-overlap length is less than a third of others. The cost of transform of $D4-D5$ -AND is around half of others'. Two optimized weights yield the longest non-overlap length and the highest costs of transform, yet their overlap length values are the same as that of the equal weight.

The detector is also applied on data with negative annotations. The specificities of negative annotations are also computed. Table 4.6 summarizes the sensitivity on paroxysmal annotations and the specificity on negative annotations.

There is a general trend that the values of specificity are inversely proportional to those of sensitivity. We can observe that in most cases in Table 4.6. For example, using the DB2 mother wavelet, when the sensitivity of the average referential is 41.61%, the specificity is 94.30%; when the sensitivity increases to 55.43%, the specificity is 86.99%. The sensitivity is increased by 13.82% with a 7.31% decrease in specificity. Table 4.6 also indicates when using Indiradevi's algorithm in YB detection, the mother wavelet DB2 yields better sensitivity results than DB4 does. When using the same weight choice, the sensitivities of DB2 are 1% to 9% better than those of DB4, while the specificities of DB2 are less than 2% worse than those of DB4.

Table 4.2: Detector results using equal weight

		RefAvg			BPAP		
		total	AEP	nonAEP	total	AEP	nonAEP
DB4	Eva#1	143.7384	90.66972	147.2377	135.6721	91.75	138.011
	Eva#2	118.5428	93.93103	118.9659	122.3696	88.57143	122.75
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	60.71453	56.06422	61.02117	65.66468	53.60294	66.30697
	Eva#5	2.606701	-	-	3.237144	-	-
	Eva#6	0.503194	0.651606	0.493408	0.569487	0.612501	0.567197
	Eva#7	107.084	74.94495	109.2033	156.9383	54.86765	162.3735
	Eva#8	81.56776	-	-	97.51155	-	-
	Eva#9	118.5428	93.93103	118.9659	122.3696	88.57143	122.75
	Eva#10	85.75534	-	-	101.5874	-	-
	Eva#11	1.532589	1.153019	1.557618	2.091251	0.873616	2.15609
	Eva#12	190.1078	109.5505	195.4198	226.9457	93.01471	234.0775
	Eva#13	81.56776	-	-	97.51155	-	-
	Eva#14	118.5428	93.93103	118.9659	122.3696	88.57143	122.75
	Eva#15	89.31986	-	-	105.0386	-	-
	Eva#16	2.029394	1.501413	2.06421	2.521764	1.261115	2.588894
	sensitivity	34.52%	65.06%	31.67%	52.83%	82.93%	49.51%
selectivity	19.30%	-	-	15.86%	-	-	
DB2	Eva#1	147.1954	90.3871	150.7175	139.1347	89.80822	141.4866
	Eva#2	117.7898	93.7619	118.1191	122.1468	102.5556	122.3208
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	61.79849	60	61.91	66.23504	57.61644	66.64598
	Eva#5	2.594635	-	-	3.177063	-	-
	Eva#6	0.517122	0.695745	0.506047	0.576178	0.670468	0.571682
	Eva#7	107.2137	65.57258	109.7955	205.1191	55.06849	212.2737
	Eva#8	80.88505	-	-	93.4005	-	-
	Eva#9	117.7898	93.7619	118.1191	122.1468	102.5556	122.3208
	Eva#10	84.25703	-	-	99.6378	-	-
	Eva#11	1.562746	1.037837	1.59529	2.770931	0.896764	2.860293
	Eva#12	192.6106	95.95968	198.603	278.0187	87.26027	287.1143
	Eva#13	80.88505	-	-	93.4005	-	-
	Eva#14	117.7898	93.7619	118.1191	122.1468	102.5556	122.3208
	Eva#15	87.84246	-	-	103.1345	-	-
	Eva#16	2.045624	1.342092	2.089243	3.194752	1.226297	3.288611
	sensitivity	41.61%	74.70%	37.93%	63.00%	89.02%	58.89%
selectivity	17.42%	-	-	14.58%	-	-	

Table 4.3: Detector results of AND decision by $D4$ & $D5$

		RefAvg			BPAP		
		total	AEP	nonAEP	total	AEP	nonAEP
DB4	Eva#1	142.5756	92.8	154.9371	137.3716	95.07895	143.6008
	Eva#2	122.1857	89.95604	122.8174	122.3734	87.86364	123.0602
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	49.91512	50.28	49.8245	54.13851	56.13158	53.84496
	Eva#5	1.753284	-	-	2.569756	-	-
	Eva#6	0.425063	0.571405	0.38872	0.455413	0.598598	0.434324
	Eva#7	27.92308	56.38667	20.8543	28.64527	62.28947	23.68992
	Eva#8	62.32586	-	-	65.44613	-	-
	Eva#9	122.1857	89.95604	122.8174	122.3734	87.86364	123.0602
	Eva#10	87.5198	-	-	84.2848	-	-
	Eva#11	0.359312	0.852532	0.236824	0.341465	0.984595	0.246741
	Eva#12	120.5836	98.90667	125.9669	111.8784	101.2368	113.4457
	Eva#13	62.32586	-	-	65.44613	-	-
	Eva#14	122.1857	89.95604	122.8174	122.3734	87.86364	123.0602
	Eva#15	90.77453	-	-	88.23557	-	-
	Eva#16	0.934249	1.281127	0.848104	0.886052	1.385997	0.812417
	sensitivity	7.39%	45.18%	5.97%	11.63%	46.34%	10.27%
selectivity	21.69%	-	-	20.27%	-	-	
DB2	Eva#1	139.1663	89.05195	150.0056	142.2216	92	149.2491
	Eva#2	122.3318	93.13483	122.898	121.6467	90.41463	122.2346
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	57.7552	58.54545	57.58427	58.25449	61.82927	57.75427
	Eva#5	2.15884	-	-	2.964199	-	-
	Eva#6	0.498639	0.678845	0.459661	0.501252	0.677255	0.476624
	Eva#7	23.44573	47.4026	18.26404	26.32934	59.90244	21.6314
	Eva#8	64.74726	-	-	68.7285	-	-
	Eva#9	122.3318	93.13483	122.898	121.6467	90.41463	122.2346
	Eva#10	86.45796	-	-	84.46039	-	-
	Eva#11	0.307839	0.778418	0.206056	0.330599	0.95286	0.243524
	Eva#12	104.8568	77.90909	110.6854	110.2964	90.07317	113.1263
	Eva#13	64.74726	-	-	68.7285	-	-
	Eva#14	122.3318	93.13483	122.898	121.6467	90.41463	122.2346
	Eva#15	89.50104	-	-	88.73294	-	-
	Eva#16	0.8092	1.099573	0.746395	0.829347	1.275605	0.766901
	sensitivity	8.48%	46.39%	7.07%	13.12%	50.00%	11.61%
selectivity	22.53%	-	-	20.59%	-	-	

Table 4.4: Detector results of optimal weight #1

		RefAvg			BPAP		
		total	AEP	nonAEP	total	AEP	nonAEP
DB4	Eva#1	146.1692	90.87903	149.0402	140.234	90.30556	142.3905
	Eva#2	116.5989	94.46512	116.9473	121.9826	97.7	122.2489
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	60.96815	56.01613	61.22529	68.18459	56.91667	68.67127
	Eva#5	2.231593	-	-	2.753222	-	-
	Eva#6	0.515109	0.65293	0.507952	0.597506	0.662158	0.594713
	Eva#7	143.3352	79.70161	146.6394	283.5118	61.34722	293.1074
	Eva#8	82.27167	-	-	97.46335	-	-
	Eva#9	116.5989	94.46512	116.9473	121.9826	97.7	122.2489
	Eva#10	85.89477	-	-	105.5007	-	-
	Eva#11	1.967418	1.199278	2.007305	3.744229	0.984602	3.863421
	Eva#12	228.5362	114.5645	234.4544	355.5612	94.73611	366.8266
	Eva#13	82.27167	-	-	97.46335	-	-
	Eva#14	116.5989	94.46512	116.9473	121.9826	97.7	122.2489
	Eva#15	89.01335	-	-	108.41	-	-
	Eva#16	2.45231	1.546348	2.499353	4.146723	1.322444	4.268708
	sensitivity	49.22%	74.10%	44.67%	68.30%	87.80%	62.99%
selectivity	13.65%	-	-	10.48%	-	-	
DB2	Eva#1	150.3565	90.77698	153.5162	144.0296	90.65333	146.2291
	Eva#2	114.9676	93.62963	115.1949	119.7274	97.14286	119.9335
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	62.01232	59.52518	62.14422	68.73509	59.28	69.12473
	Eva#5	2.267932	-	-	2.775178	-	-
	Eva#6	0.517929	0.69151	0.508724	0.59336	0.689996	0.589378
	Eva#7	128.4123	64.61151	131.7959	230.7995	61.89333	237.7599
	Eva#8	79.68452	-	-	93.819	-	-
	Eva#9	114.9676	93.62963	115.1949	119.7274	97.14286	119.9335
	Eva#10	82.66395	-	-	99.77684	-	-
	Eva#11	1.746226	1.021636	1.784654	3.111634	1.003158	3.198521
	Eva#12	216.7565	95.86331	223.1679	306.0939	93.26667	314.8643
	Eva#13	79.68452	-	-	93.819	-	-
	Eva#14	114.9676	93.62963	115.1949	119.7274	97.14286	119.9335
	Eva#15	85.89489	-	-	102.8169	-	-
	Eva#16	2.228297	1.330126	2.27593	3.518274	1.313163	3.609144
	sensitivity	54.08%	83.73%	48.68%	74.43%	91.46%	68.87%
selectivity	13.20%	-	-	10.28%	-	-	

Table 4.5: Detector results of optimal weight #2

		RefAvg			BPAP		
		total	AEP	nonAEP	total	AEP	nonAEP
DB4	Eva#1	144.2023	90.87903	146.955	139.7237	90.41096	141.8678
	Eva#2	116.8324	94.46512	117.1877	122.4253	97.66667	122.6726
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	63.07482	56.04032	63.43797	69.50457	56.56164	70.0673
	Eva#5	2.351031	-	-	2.888536	-	-
	Eva#6	0.533512	0.653097	0.527338	0.610646	0.657901	0.608592
	Eva#7	170.2308	79.75	174.9017	341.2511	60.50685	353.4574
	Eva#8	87.09047	-	-	103.128	-	-
	Eva#9	116.8324	94.46512	117.1877	122.4253	97.66667	122.6726
	Eva#10	91.39632	-	-	113.4407	-	-
	Eva#11	2.323478	1.199743	2.381489	4.744983	0.971115	4.909065
	Eva#12	251.3583	114.5887	258.4188	411.4703	94.35616	425.2579
	Eva#13	87.09047	-	-	103.128	-	-
	Eva#14	116.8324	94.46512	117.1877	122.4253	97.66667	122.6726
	Eva#15	94.42024	-	-	116.3589	-	-
	Eva#16	2.789966	1.546647	2.854151	5.134337	1.313213	5.300473
	sensitivity	49.49%	74.10%	45.18%	68.81%	89.02%	63.43%
selectivity	13.67%	-	-	10.56%	-	-	
DB2	Eva#1	147.5023	90.77698	150.4335	141.7035	90.65333	143.7833
	Eva#2	115.499	93.62963	115.7398	121.3008	97.14286	121.5321
	Eva#3	123.3557	91.24096	124.4311	123.6277	91.20732	124.7066
	Eva#4	64.78685	59.68345	65.05056	72.16388	59.24	72.69039
	Eva#5	2.454856	-	-	3.010101	-	-
	Eva#6	0.544579	0.693032	0.536908	0.625277	0.689448	0.622662
	Eva#7	184.9502	69.64029	190.9086	321.7323	64.77333	332.2004
	Eva#8	87.30179	-	-	103.4834	-	-
	Eva#9	115.499	93.62963	115.7398	121.3008	97.14286	121.5321
	Eva#10	91.93805	-	-	112.8733	-	-
	Eva#11	2.484295	1.073579	2.55719	4.543657	1.030328	4.686786
	Eva#12	267.6656	100.7338	276.2914	391.2719	96.18667	403.2933
	Eva#13	87.30179	-	-	103.4834	-	-
	Eva#14	115.499	93.62963	115.7398	121.3008	97.14286	121.5321
	Eva#15	95.07225	-	-	115.7739	-	-
	Eva#16	2.939716	1.380547	3.020282	4.918381	1.340881	5.064123
	sensitivity	55.43%	83.73%	50.34%	75.26%	91.46%	70.33%
selectivity	13.29%	-	-	10.27%	-	-	

Table 4.6: Summary of Indiradevi's algorithm

		sensitivity of AEP (bipolar)	sensitivity (bipolar)	specificity (avg ref)	sensitivity of AEP (avg ref)	sensitivity (avg ref)
equal weights	DB4	82.93%	52.83%	95.96%	65.06%	34.52%
	DB2	89.02%	63.00%	94.30%	74.70%	41.61%
<i>D4-D5-AND</i>	DB4	46.34%	11.63%	99.37%	45.18%	7.39%
	DB2	50.00%	13.12%	99.53%	46.39%	8.48%
optimal#1	DB4	87.80%	68.30%	89.56%	74.10%	49.22%
	DB2	91.46%	74.43%	87.79%	83.73%	54.08%
optimal#2	DB4	89.02%	68.81%	89.26%	74.10%	49.49%
	DB2	91.46%	75.26%	86.99%	83.73%	55.43%

4.2.2 Implementation of ANN

Neural net is highly sensitive to its training parameters. In this section, a hidden layer with 9 units is used; the maximum training epoch is set as 50; 5-level wavelet decomposition is implemented using mother wavelet DB2; and the feature set choice is Set #1 in Section 3.1.1.2. The evaluation method is the same as it is in Section 4.2.1.

Multiple tests were performed with different ratios of training data and different output units. The details are listed below:

Test#1: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:10 (in this case, the ratio of YB:negative is 2:10);

Test#2: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:8 (the ratio of YB:negative is 2:8);

Test#3: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:6 (the ratio of YB:negative is 2:6);

Test#4: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:4 (the ratio of YB:negative is 2:4);

Test#5: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:2 (the ratio of YB:negative is 2:2);

Test#6: The output layer unit number is 3; The ratio of AEP:nonAEP:negative is 1:1:1 (the ratio of YB:negative is 2:1);

- Test#7:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:10 (the ratio of YB:negative is 2:10);
- Test#8:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:8 (the ratio of YB:negative is 2:8);
- Test#9:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:6 (the ratio of YB:negative is 2:6);
- Test#10:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:4 (the ratio of YB:negative is 2:4);
- Test#11:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:2 (the ratio of YB:negative is 2:2);
- Test#12:** The output layer unit number is 2; The ratio of AEP:nonAEP:negative is 1:1:1 (the ratio of YB:negative is 2:1); and
- Test#13:** The output layer unit number is 2; The ratio of YB:negative is 1:1 (bind AEP and non-AEP).

Table 4.7 summarizes the performances of the 13 tests. We observed that the sensitivity of AEP is still fine while the specificity drops rapidly when the proportion of negative data is reduced in the training data. Figure 4.3 illustrates the trends of sensitivity and specificity when the ratio is changing. According to Figure 4.3, the equal error rate of YB detection is 75% and that of ETs is 88% .

4.2.3 Interpretation of S_{ij} and Implementation of ANN with Pruning Strategy

4.2.3.1 S_{ij} of Input Layer's Weights

The test was performed on a small scale dataset, which includes feature vectors obtained from both paroxysmal and negative YBs. As in classification stage, each YB yields a single vector with selected wavelet (DB2 in this case). We train the net for 6000 epochs. The objective is to detect all the ETs and as many non-ETs as possible with the minimal occurrence of false positives. To avoid

Table 4.7: ANN performances in YB detection

	ratio	sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	
Test#1	1:1:10	55.54%	86.08%	54.56%	88.29%	3 outputs
Test#2	1:1:8	56.37%	88.61%	55.33%	88.16%	
Test#3	1:1:6	65.25%	92.41%	64.37%	83.57%	
Test#4	1:1:4	71.95%	91.14%	71.34%	79.03%	
Test#5	1:1:2	81.74%	100.00%	81.15%	67.33%	
Test#6	1:1:1	90.93%	98.75%	90.68%	49.14%	
Test#7	1:1:10	62.42%	92.50%	61.44%	85.16%	2 outputs
Test#8	1:1:8	67.65%	93.67%	66.82%	81.29%	
Test#9	1:1:6	73.41%	97.47%	72.64%	75.32%	
Test#10	1:1:4	80.36%	96.20%	79.85%	68.71%	
Test#11	1:1:2	90.34%	98.73%	90.07%	52.14%	
Test#12	1:1:1	96.53%	100.00%	96.42%	32.43%	
Test13	1YB:1neg.	93.93%	100.00%	93.72%	50.29%	

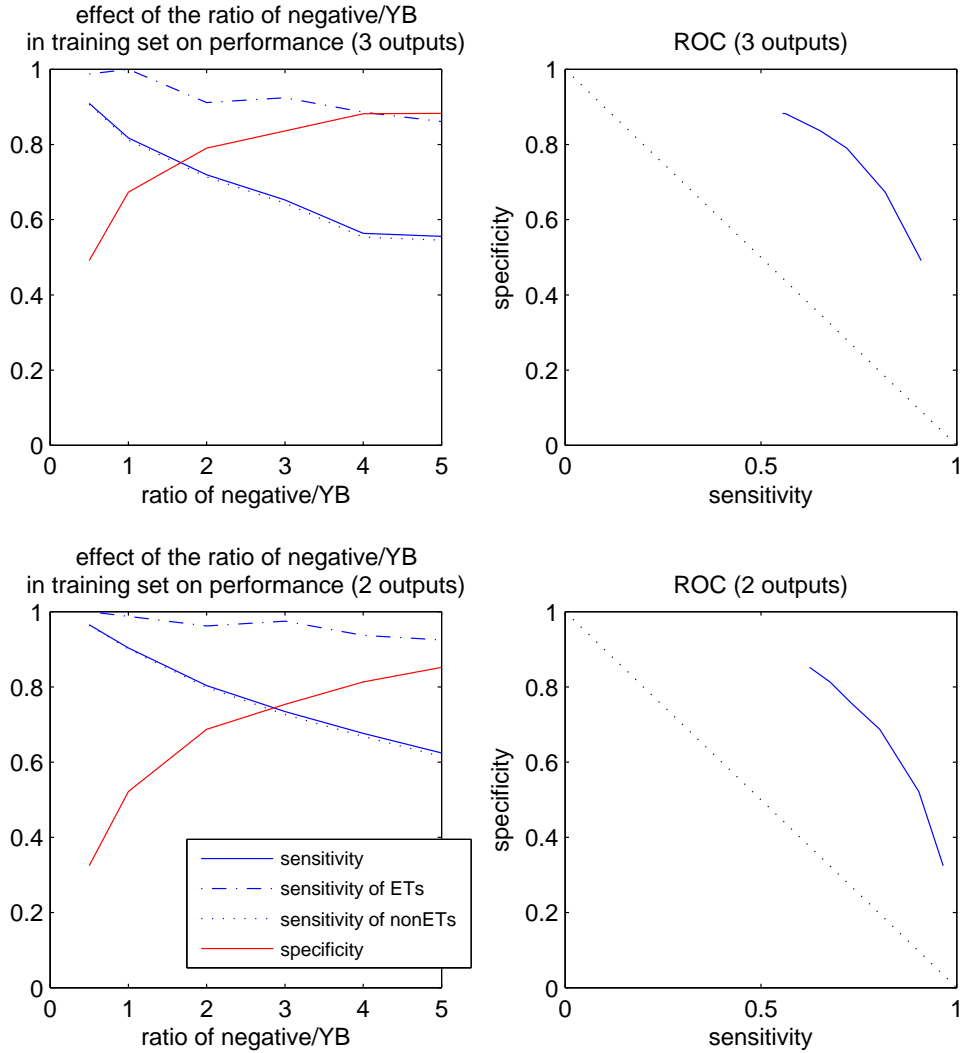
partiality to either paroxysmal or negative class during learning, we balance the training dataset as *paroxysmal/negative* = 1 : 1, and *ETs/non-ETs* = 1 : 1 in paroxysmal class. The training data are randomly selected. We set the initial learning rate as $\eta_a = 1e-6$ and $\mu = 0.001$. The learning rate is changed every 1000 epoch. All the weights are randomly initialized with numerical values less than $1e-4$. Bias and momentum are eliminated. The objective output of paroxysmal events is set as $[1 \ -1]^T$ and that of negative events is $[-1 \ 1]^T$.

The learning rates developed and used respectively in the six intervals are $1e-6$, $3.73e-08$, $4.10e-07$, $4.71e-08$, $1.19e-07$, and $3.34e-08$. The S_{ij} of each weight is computed using the algorithm in Section 4.1.2.1. Every input feature is directly connected to each of the 51 hidden layer units. Thus, there are 1275 total input weights and corresponding S_{ij} estimates. Instead of inspecting individual S_{ij} values, we partition the S_{ij} related to one feature or one set of features by their value ranges in each decomposition level. Three ranges of S_{ij} are used, as shown in the legend of Figure 4.4. Furthermore, three categories of inputs are considered:

1. S_{ij} related to individual input (top section of Figure 4.4);
2. S_{ij} related to individual subband (middle section of Figure 4.4); and
3. S_{ij} related to individual feature type, independent of subband (bottom section of Figure 4.4).

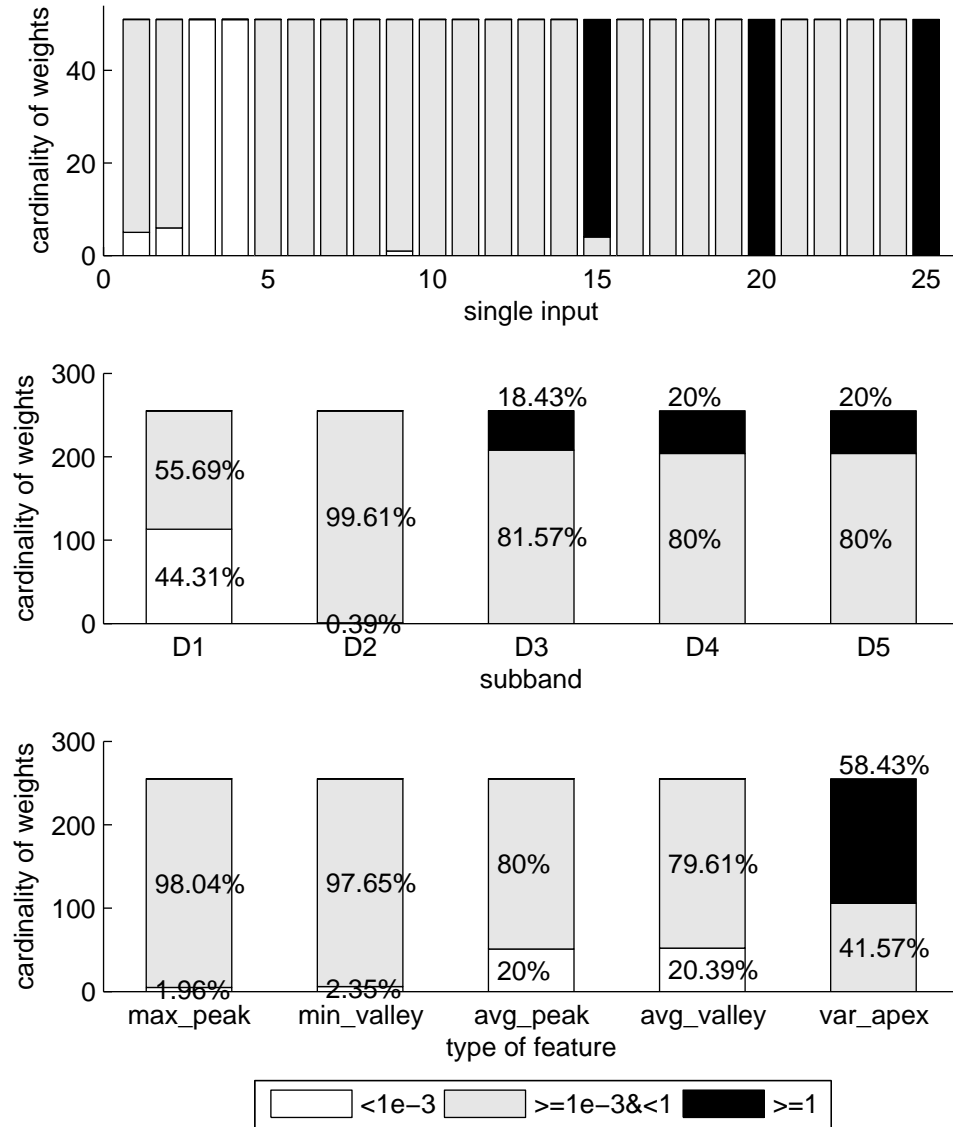
From the middle section of Figure 4.4, we observe that 44.31% of the S_{ij} related to the input features of subband D1 (white region of the first column in the middle section of Figure 4.4) are less than 0.001, indicating the removal of nearly half of the weights related to subband D1 would

Figure 4.3: ANN performance with different ratio of training data



hardly affect the output error. In subband D3, D4 and D5, the proportion of the S_{ij} larger than 1 approximates to 20% (black region in the middle section of Figure 4.4). Notice that the frequency ranges of these three subbands have overlaps with the frequency bands of spikes and sharp waves of ETs. We also observe that 20% of the S_{ij} related to two features, ‘mean of the peaks’ and ‘mean of the valleys’ (white annotation of the fourth and fifth columns in the bottom section of Figure 4.4), are less than 0.001. This suggests inputs related to the two ‘mean’ features are less significant than those related to other features. It is especially noteworthy that all the S_{ij} larger than 1 are related

Figure 4.4: The proportion of values of S_{ij} for selected input



to the feature ‘the variance of the peaks and the valleys’, which indicates a strong relation between this feature and the error function.

4.2.3.2 Confirmation of Results with Feature/Subband Pruning

The results in Section 4.2.3.1 show the estimated relation between subbands/features and the ANN mapping error function. They serve as an indirect measure of the utility of the various features and subbands. To verify the results experimentally, we use the same network and parameters but remove low S_{ij} inputs (subband or feature) from the input. We also reduce the training epoch to 3000. 10-fold cross validation is used to evaluate this mapping performance. The results are shown in Table 4.8.

Table 4.8: Performance of ANN with restricted input

	sensitivity of ETs difference to (*)		sensitivity of non-ETs difference to (*)		specificity difference to (*)	
all subband(*)	92.50%	-	73.91%	-	74.88%	-
without D1	92.50%	(0.00%)	73.91%	(0.00%)	75.24%	(0.37%)
without D2	92.50%	(0.00%)	68.71%	(-5.20%)	76.71%	(1.83%)
without D3	93.75%	(1.25%)	77.30%	(3.39%)	69.27%	(-5.61%)
without D4	87.50%	(-5.00%)	74.64%	(0.73%)	65.37%	(-9.51%)
without D5	91.25%	(-1.25%)	70.60%	(-3.31%)	76.95%	(2.07%)
without $\max\{p\}$	91.25%	(-1.25%)	71.57%	(-2.34%)	77.68%	(2.80%)
without $\min\{v\}$	83.75%	(-8.75%)	67.98%	(-5.93%)	77.32%	(2.44%)
without \bar{p}	95.00%	(2.50%)	76.33%	(2.42%)	71.10%	(-3.78%)
without \bar{v}	83.75%	(-8.75%)	68.47%	(-5.44%)	79.15%	(4.27%)
without $\text{var}\{p,v\}$	95.00%	(2.50%)	63.55%	(-10.36%)	84.88%	(10.00%)

In Table 4.8, we notice that when the subband D1 is removed, the sensitivity of ETs and the sensitivity of non-ETs remain unchanged while the change of the specificity is only 0.37%. However, the removal of subband D4 results in 5% decrease in sensitivity of ETs and 9.51% decrease in specificity. The removal of other subbands also leads to obvious changes in sensitivities and specificities. When increase and decrease occur in sensitivity and specificity, respectively, in the same case, the extent of decrease is larger than that of increase. Despite the fact that the two features, ‘mean of the peaks’ and ‘mean of the valleys’, have a similar proportion of small S_{ij} , their performances are quite different: when the feature ‘mean of the peaks’ is removed, the largest absolute change in performance is only 3.78%; when the feature ‘mean of the valleys’ is removed, the sensitivity of ETs drops 8.75% and the sensitivity of non-ETs drops 5.44%. The removal of ‘the variance’ causes 10% decrease in the sensitivity of non-ETs but same rate of increase in the

specificity.

Comparing the results of proportions of S_{ij} values with different feature/subband choices in Section 4.2.3.1 and the performance in Section 4.2.3.2, we conclude that the subband D1, although useful in paroxysmal classification [13], is insignificant in YBs detection. Although all the other four detail subbands have overlaps with the frequency ranges of ETs, only subband D4 affects the overall performances. D4 is thus considered as the most important subband. The removal of D2, D3 and D5 causes significant decreases in either sensitivity or specificity.

There is no clear evidence, either from the estimated S_{ij} or the experimental verification, about any preferences for individual features. When either of the features ‘highest peak’, ‘lowest valley’, ‘mean of the valleys’ or ‘the variance’ is removed, the neural net tended to favor the negative YBs in some degree. Conversely, the removal of the feature ‘mean of the peaks’ favors paroxysmal YBs. The coexistence of increase and decrease in performance indicates that the characteristics of paroxysmal YBs and those of negative YBs do not distribute evenly in one type of features.

Since the removal of subband D1 barely affects the performance, we re-tested the data with wavelet DB4 and the combination DB4+DB2 under the condition that features from D1 are removed. Comparison of the performances are shown in Table 4.9. The performance of DB2 is slightly better than that of DB4. Notice that the employment of dual mother wavelets improves the performance significantly again: The sensitivity of ETs and specificity are over 95% with a relatively high sensitivity of non-ETs at 88%.

Table 4.9: Comparison of the ANN performances with features from subband D1 eliminated

wavelet choice	sensitivity	sensitivity of ETs	sensitivity of non-ETs	specificity
DB2	74.49%	92.50%	73.91%	75.24%
DB4	73.87%	92.50%	73.27%	73.90%
DB4+DB2	88.13%	95.00%	87.90%	97.20%

4.2.3.3 Re-Test of ANN after Pruning

Section 4.2.3.1 discusses the possibility of reducing input data dimension by pruning and Section 4.2.3.2 confirms it through small scale tests. To compare the performances with those in Section 4.2.2, we test on the real-time recordings, using the three neural nets yielding the results in Table 4.9. The output layer unit number is 2; the ratio of AEP:nonAEP:negative is 1:1:2 (the ratio of YB:negative is 2:2). Below are details of each test:

Test#14: The neural net is trained by features yielded from DB2 wavelet in Table 4.9;

Test#15: The neural net is trained by features yielded from DB4 wavelet in Table 4.9; and

Test#16: The neural net is trained by features yielded from DB4 and DB2 wavelet in Table 4.9.

Table 4.10: ANN performance in YB detections after pruning of input features

	sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	wavelet choice
Test#14	96.41%	100.00%	96.29%	31.10%	DB2
Test#15	96.61%	100.00%	96.50%	31.10%	DB4
Test#16	99.96%	100.00%	99.96%	1.28%	DB4+DB2

The results are shown in Table 4.10. All the sensitivity results are outstanding in Table 4.10, though the specificity results are poor. All tests yield high sensitivities of AEP. Compared to Test#11 in Table 4.7, the sensitivity of nonAEP is improved by 6% to 10%, while the specificity drops 21% to 51%.

4.2.4 Implementation of ANN with Morphological Features

As mentioned in Section 4.1.2.2, ten morphology based features are tested with neural net as reference. The features are extracted from raw EEG recordings. Below are details of each test:

Test#17: The ratio of AEP:AP:NEP:negative is 1:1:1:3 (the ratio of YB:negative is 3:3 while that of ET:nonET is 1:2);

Test#18: The ratio of AEP:AP:NEP:negative is 2:1:1:4 (the ratio of YB:negative is 4:4 while that of ET:nonET is 2:2).

Table 4.11: ANN performance in YB detections using morphological features

	ratio	sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	
Test#17	1:1:1:3	46.07%	59.49%	45.63%	91.70%	4 outputs
Test#18	2:1:1:4	62.57%	88.61%	61.73%	79.63%	

Test#18 achieves a moderate sensitivity of AEP and specificity, although the sensitivity of nonAEP is not impressive. Judging from the values, Test#18 is close to Test#8 in Section 4.2.2, with relatively lower sensitivity of AEP and specificity.

4.2.5 Performance of Full-scale Real-time Simulation

The full-scale simulation is accomplished with ANN. The patients are randomly divided into 10 folds with the restriction in Section 4.1.4 satisfied. Train epoch number is 3000; initial learning rate is $1e-6$ and will be adjusted in every 300 epochs; number of hidden layer units is 41; number of output layer units is 2 while the objective output of paroxysmal events is $[1 \ -1]^T$ and that of negative events is $[-1 \ 1]^T$. All the weights are randomly initialized with numerical values less than $1e-4$; momentum and bias are banned.

The threshold of outlier discarding is 5; in each trial, two output candidates are merged if the interval between them is equal or less than threshold: (a) 0, (b) 18, and (c) 52; also two overlap rates of grouping are chosen: 50% and $1e-3\%$; notice that since the length of EEG segment is 7680, one sample is $1.3e-2\%$ of the segment; the choice of $1e-3\%$ is short enough to guarantee grouping as long as there is overlap.

In the first round, wavelet features created by DB4 are used, with subband D1 ruled out. The ratio of class in each simulation is listed below:

Simulation#1: The ratio of AEP:nonAEP:negative is 1:1:10 (the ratio of YB:negative is 2:10 while that of ET:nonET is 1:1);

Simulation#2: The ratio of AEP:nonAEP:negative is 1:2:10 (the ratio of YB:negative is 3:10 while that of ET:nonET is 1:2);

Simulation#3: The ratio of AEP:nonAEP:negative is 1:1:5 (the ratio of YB:negative is 2:5 while that of ET:nonET is 1:1);

Simulation#4: The ratio of YB:negative is 1:1 while that of ET:nonET is unknown;

Simulation#5: The ratio of YB:negative is 1:5 while that of ET:nonET is unknown;

Simulation#6: The ratio of AEP:nonAEP:negative is 1:0:1 (the ratio of YB:negative is 1:1 while that of ET:nonET is 1:0);

Simulation#7: The ratio of AEP:nonAEP:negative is 1:0:5 (the ratio of YB:negative is 1:5 while that of ET:nonET is 1:0).

In Table 4.12, we notice that the sensitivity and specificity values stay the same when the candidates are merged with different thresholds. However, the selectivity increases significantly when

Table 4.12: Simulation results without grouping

trial		sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	selectivity
Simulation#1	(a)	25.26%	32.93%	25.00%	96.36%	4.67%
	(b)	25.26%	32.93%	25.00%	96.36%	12.95%
	(c)	25.26%	32.93%	25.00%	96.36%	16.89%
Simulation#2	(a)	47.37%	70.73%	46.59%	88.36%	3.73%
	(b)	47.37%	70.73%	46.59%	88.36%	9.77%
	(c)	47.45%	70.73%	46.67%	88.36%	13.05%
Simulation#3	(a)	25.41%	60.98%	24.23%	91.15%	3.57%
	(b)	25.41%	60.98%	24.23%	91.15%	8.11%
	(c)	25.41%	60.98%	24.23%	91.15%	10.43%
Simulation#4	(a)	88.92%	100.00%	88.56%	56.36%	1.99%
	(b)	88.92%	100.00%	88.56%	56.36%	5.89%
	(c)	88.96%	100.00%	88.60%	56.36%	9.34%
Simulation#5	(a)	27.10%	21.95%	27.27%	96.73%	5.14%
	(b)	27.10%	21.95%	27.27%	96.73%	12.05%
	(c)	27.18%	21.95%	27.35%	96.73%	17.00%
Simulation#6	(a)	48.27%	92.68%	46.79%	77.82%	2.66%
	(b)	48.27%	92.68%	46.79%	77.82%	5.90%
	(c)	48.27%	92.68%	46.79%	77.82%	8.15%
Simulation#7	(a)	21.29%	41.46%	20.62%	93.45%	3.01%
	(b)	21.29%	41.46%	20.62%	93.45%	6.91%
	(c)	21.29%	41.46%	20.62%	93.45%	9.33%

Table 4.13: Simulation results grouped with overlap rates of 50%

		sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	selectivity
Simulation#1	(a)	21.45%	28.05%	21.23%	96.97%	6.72%
	(b)	18.97%	23.17%	18.83%	96.97%	13.93%
	(c)	19.29%	21.95%	19.20%	96.85%	18.93%
Simulation#2	(a)	40.10%	56.10%	39.57%	90.42%	5.26%
	(b)	35.74%	43.90%	35.47%	90.55%	10.76%
	(c)	34.41%	51.22%	33.85%	92.48%	14.12%
Simulation#3	(a)	22.07%	46.34%	21.27%	92.85%	4.98%
	(b)	20.19%	36.59%	19.64%	93.82%	9.34%
	(c)	19.84%	37.80%	19.24%	94.42%	11.98%
Simulation#4	(a)	69.56%	57.32%	69.97%	63.64%	2.95%
	(b)	61.31%	47.56%	61.77%	67.03%	6.25%
	(c)	58.25%	48.78%	58.56%	70.91%	9.61%
Simulation#5	(a)	24.23%	17.07%	24.47%	97.09%	6.56%
	(b)	21.41%	12.20%	21.71%	97.21%	12.67%
	(c)	20.31%	10.98%	20.62%	97.45%	17.59%
Simulation#6	(a)	40.77%	62.20%	40.06%	80.48%	3.66%
	(b)	36.61%	51.22%	36.12%	83.52%	6.84%
	(c)	36.53%	47.56%	36.16%	85.33%	9.38%
Simulation#7	(a)	18.54%	28.05%	18.22%	94.42%	4.51%
	(b)	17.01%	26.83%	16.68%	95.03%	8.18%
	(c)	16.42%	21.95%	16.23%	95.52%	10.70%

Table 4.14: Simulation results grouped with overlap rates of 1e-3%

		sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	selectivity
Simulation#1	(a)	20.03%	20.73%	20.01%	97.45%	10.05%
	(b)	16.61%	17.07%	16.60%	97.70%	21.59%
	(c)	17.01%	13.41%	17.13%	97.58%	29.58%
Simulation#2	(a)	37.63%	53.66%	37.09%	92.37%	8.56%
	(b)	30.32%	42.68%	29.91%	93.70%	17.01%
	(c)	28.00%	39.02%	27.64%	94.91%	22.09%
Simulation#3	(a)	20.42%	41.46%	19.72%	93.94%	7.16%
	(b)	16.54%	30.49%	16.07%	95.76%	13.18%
	(c)	15.51%	32.93%	14.94%	96.61%	16.55%
Simulation#4	(a)	68.66%	51.22%	69.24%	72.24%	6.47%
	(b)	56.05%	46.34%	56.37%	76.73%	13.11%
	(c)	51.30%	35.37%	51.83%	80.00%	19.30%
Simulation#5	(a)	22.23%	9.76%	22.65%	97.45%	9.54%
	(b)	18.66%	8.54%	18.99%	97.33%	18.40%
	(c)	16.54%	8.54%	16.80%	97.94%	24.24%
Simulation#6	(a)	37.35%	54.88%	36.77%	85.33%	6.27%
	(b)	30.75%	41.46%	30.40%	89.33%	11.19%
	(c)	28.40%	32.93%	28.25%	89.82%	14.54%
Simulation#7	(a)	16.85%	26.83%	16.52%	95.52%	6.66%
	(b)	13.79%	17.07%	13.68%	96.48%	11.90%
	(c)	12.77%	17.07%	12.62%	97.09%	15.36%

Table 4.15: Morphology feature based simulation results without grouping

		sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	selectivity
Simulation#8	(a)	1.96%	9.76%	1.70%	100.00%	58.14%
	(b)	1.96%	9.76%	1.70%	100.00%	58.82%
	(c)	1.96%	9.76%	1.70%	100.00%	59.52%
Simulation9	(a)	17.99%	53.66%	16.80%	96.97%	14.43%
	(b)	17.99%	53.66%	16.80%	96.97%	14.71%
	(c)	17.99%	53.66%	16.80%	96.97%	15.29%

Table 4.16: Morphology feature based simulation results grouped with overlap rates of group50%

		sensitivity	sensitivity of AEP	sensitivity of nonAEP	specificity	selectivity
Simulation#8	(a)	1.85%	7.32%	1.66%	100.00%	63.51%
	(b)	1.85%	7.32%	1.66%	100.00%	63.51%
	(c)	1.85%	7.32%	1.66%	100.00%	64.38%
Simulation9	(a)	15.12%	42.68%	14.20%	97.82%	17.57%
	(b)	15.24%	43.90%	14.29%	97.82%	17.77%
	(c)	15.40%	43.90%	14.45%	97.82%	18.30%

the merging threshold goes up. In all cases, after merging with threshold 18, selectivity is more than twice of that before merging; after merging with threshold 52, selectivity is about three times of that before merging.

In Table 4.13 and Table 4.14, sensitivity drops in some degree while the selectivity does

not goes up as fast as in cases without grouping. Notice that the merging process is ahead of the grouping process. Merging changes the length of the candidates and thus leads to different grouping outcomes. Corresponding sensitivity values are also changed. Notice that in ‘Simulation#2(c)’, the sensitivity is 70.73% before grouping; it is reduced to 39.02% after grouping. This phenomenon indicates during the grouping process of TP, only 55.17% of the machine decisions are the same as the experts’.

In the second round, morphology features in Section 4.1.2.2 are used. The ratio of class in each simulation is listed below:

Simulation#8: with implementation of morphology feature set in Section 4.1.2.2, the ratio of AEP:nonAEP:negative is 1:2:10 (the ratio of YB:negative is 3:10 while that of ET:nonET is 1:2);

Simulation#9: with implementation of morphology feature set in Section 4.1.2.2, the ratio of AEP:nonAEP:negative is 1:1:5 (the ratio of YB:negative is 2:5 while that of ET:nonET is 1:1);

The same trend in Table 4.12, Table 4.13 and Table 4.14 also appears in Table 4.15 and 4.16. As compared to wavelet features, morphology features yielded high specificity and selectivity values. The sensitivity, on the other hand, is not impressive.

Figure 4.5 to Figure 4.14 demonstrate how the expert and the machine annotated YBs on EEGnet. Detectors based on wavelet features (Figure 4.6, Figure 4.7, Figure 4.11 and Figure 4.12) are very sensitive to different kinds of spikes and bursts. They also tend to annotate an entire event.

Detectors based on wavelet features (Figure 4.8, Figure 4.9, Figure 4.13 and Figure 4.14) tend to mark single spikes and show less interest in bursts.

Detectors based on both features missed the smoothly events of patient#5.

Comparing to experts’ marked YBs, the machine tends to favor paroxysmal events containing high frequency components. The machine cannot determine whether two close events are related while experts can determine it by reading context. The machine marks every signal pieces that fits the description while experts do the job selectively. In the grouping procedure, the machine makes the decision based on all the values in the YB, while experts focus on representative events in the YB.

Figure 4.5: Yellow boxes annotated by expert on Patient#1

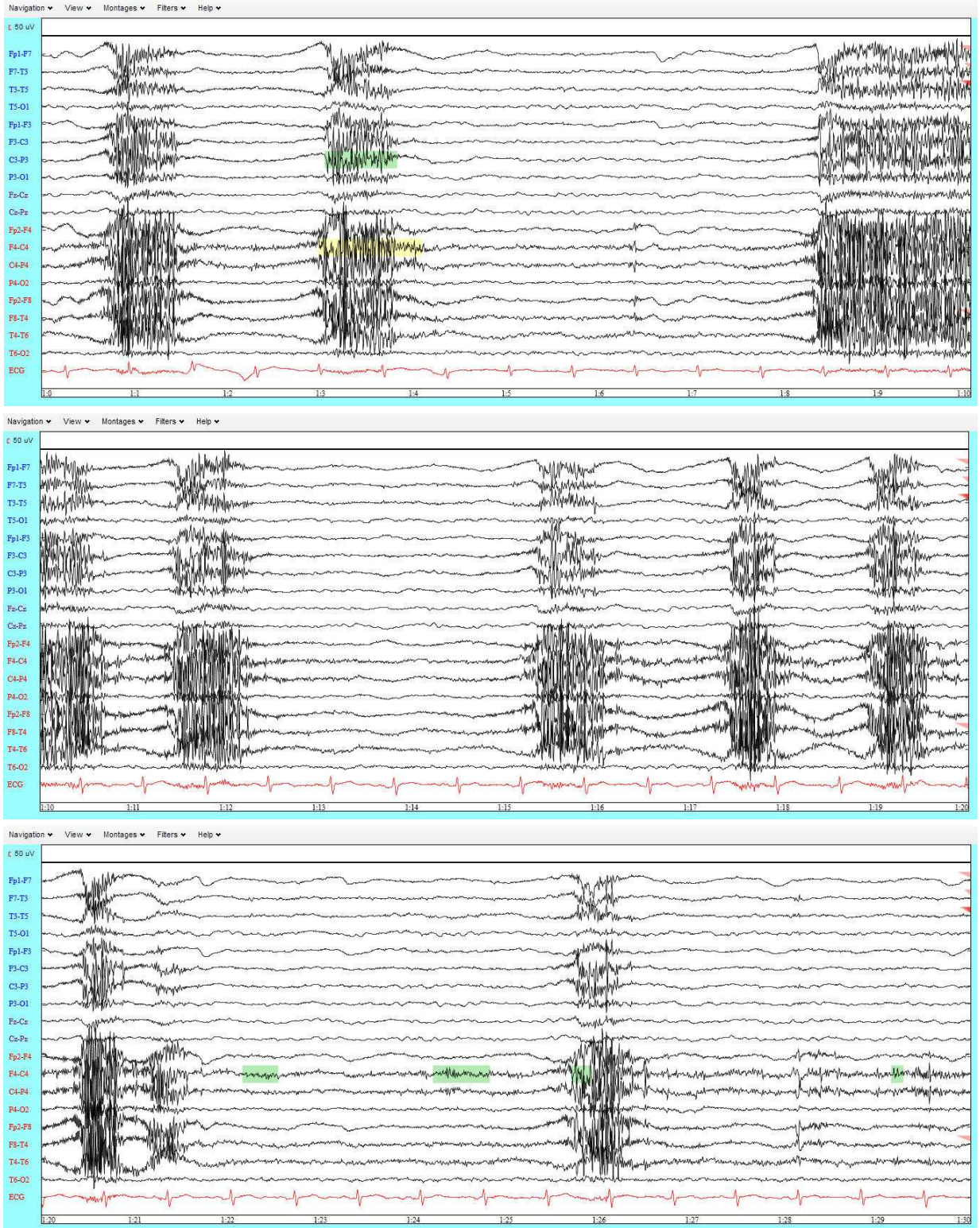


Figure 4.6: Raw yellow box candidates annotated by Simulation#2 on Patient#1

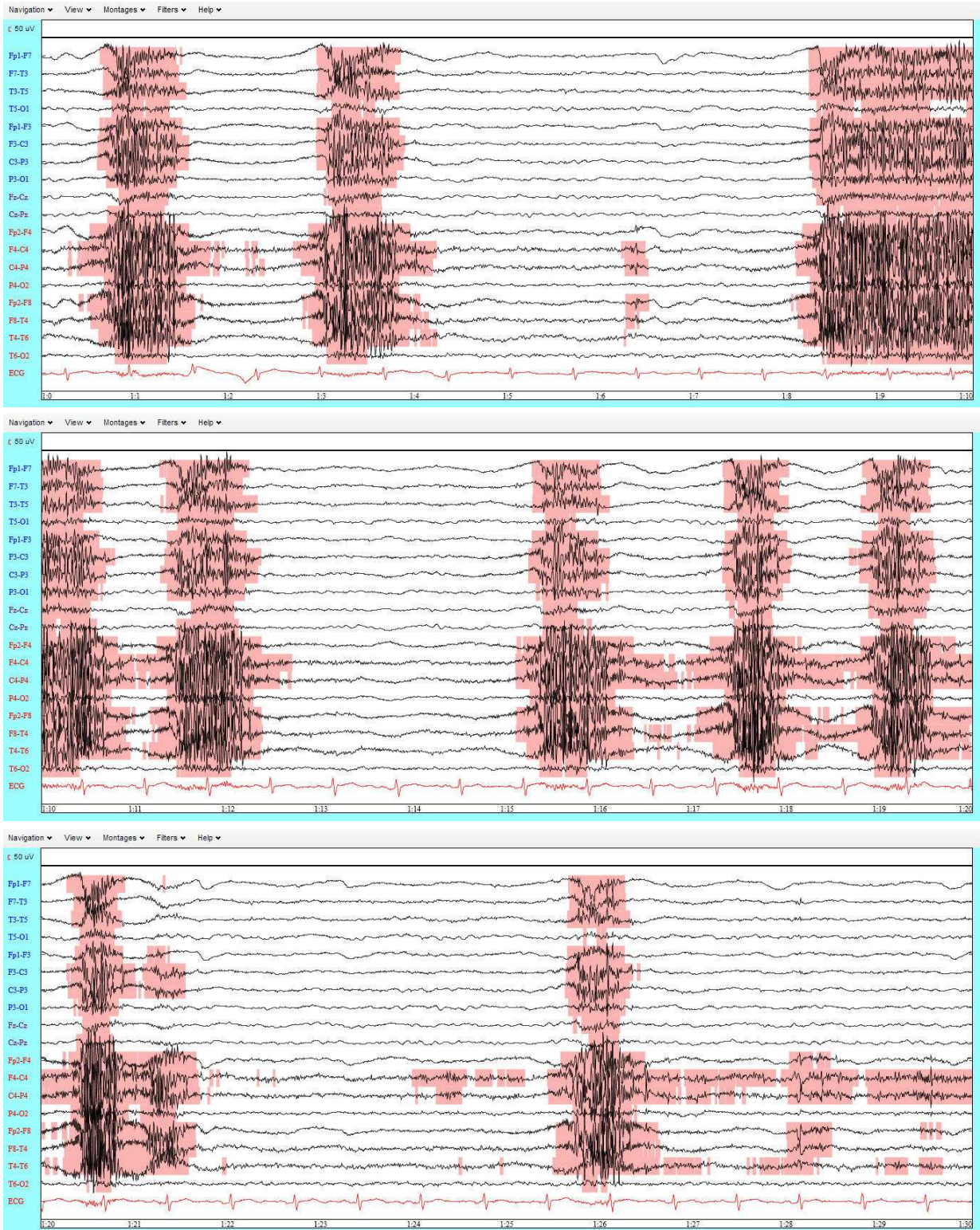


Figure 4.7: Yellow box annotated by Simulation#2 on Patient#1 after grouping

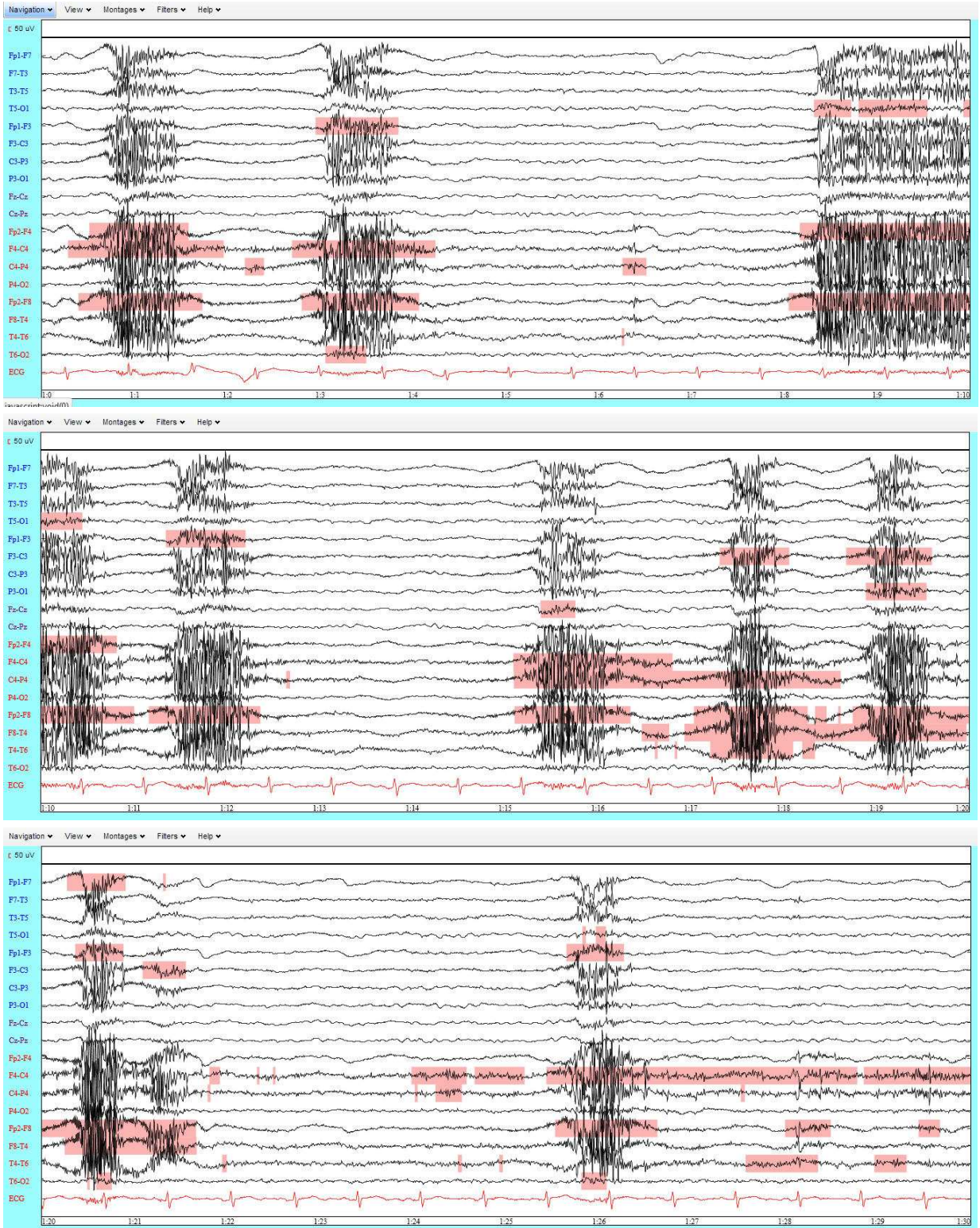


Figure 4.8: Raw yellow box candidates annotated by Simulation#9 on Patient#1

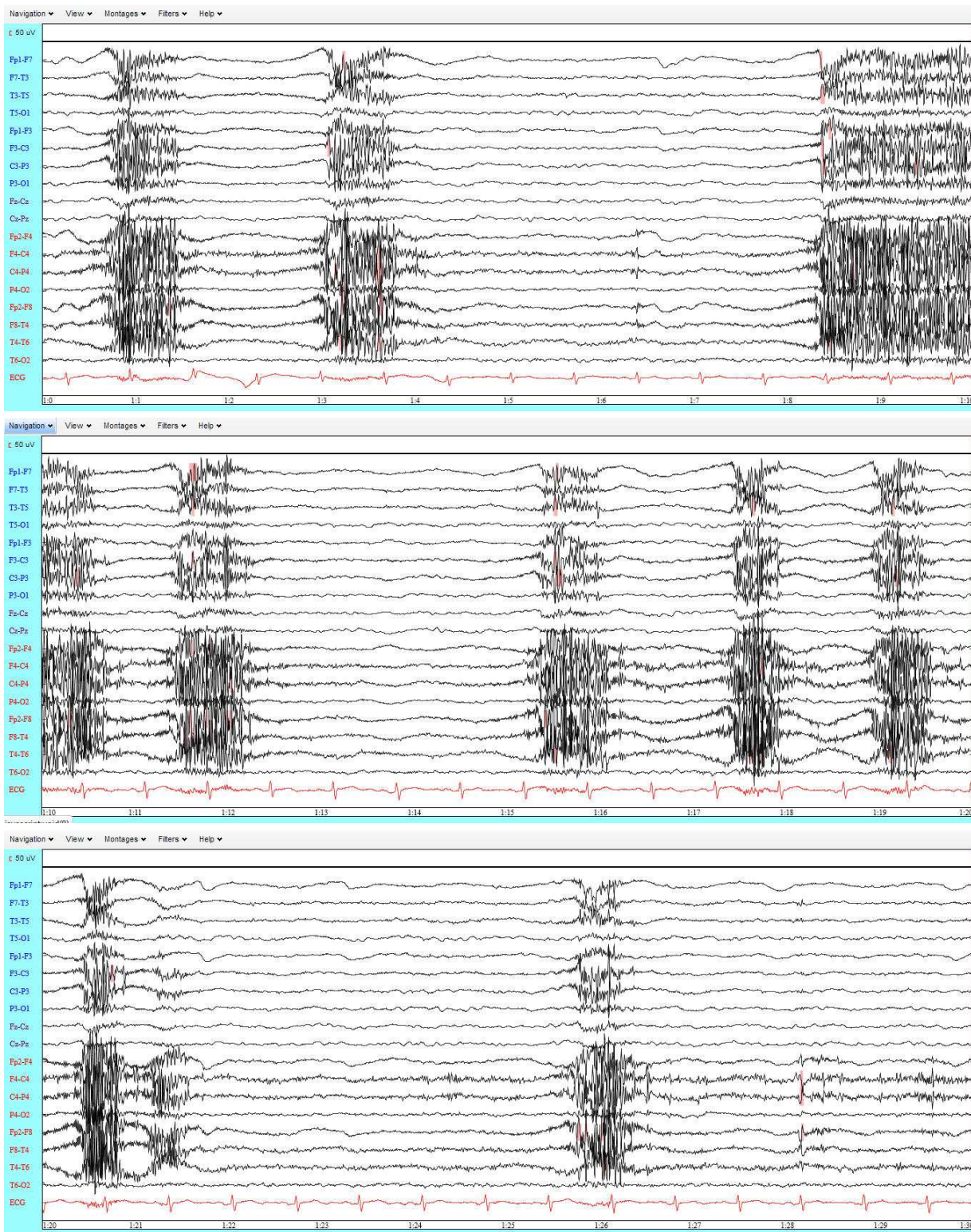


Figure 4.9: Yellow box annotated by Simulation#9 on Patient#1 after grouping

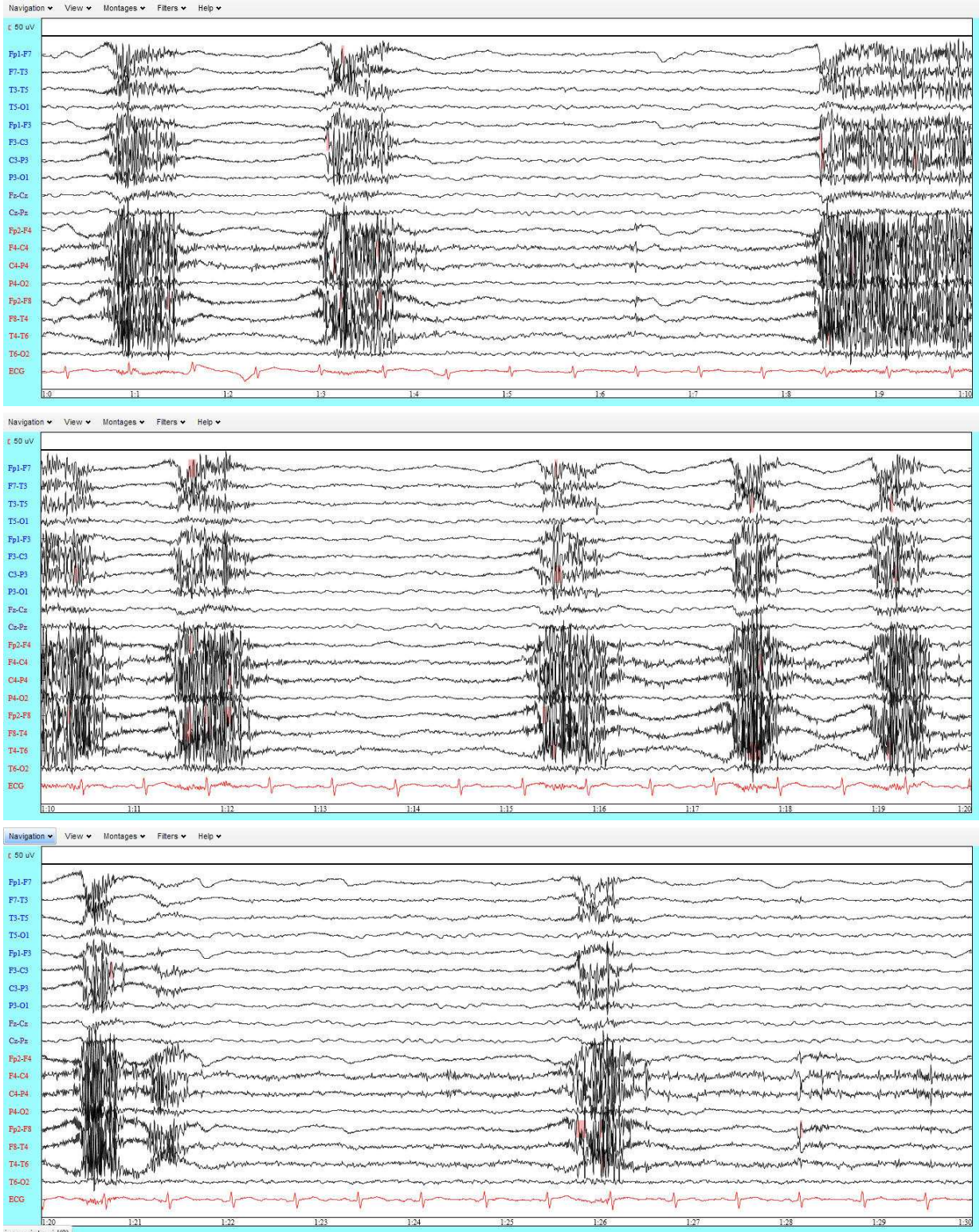


Figure 4.10: Yellow boxes annotated by expert on Patient#5

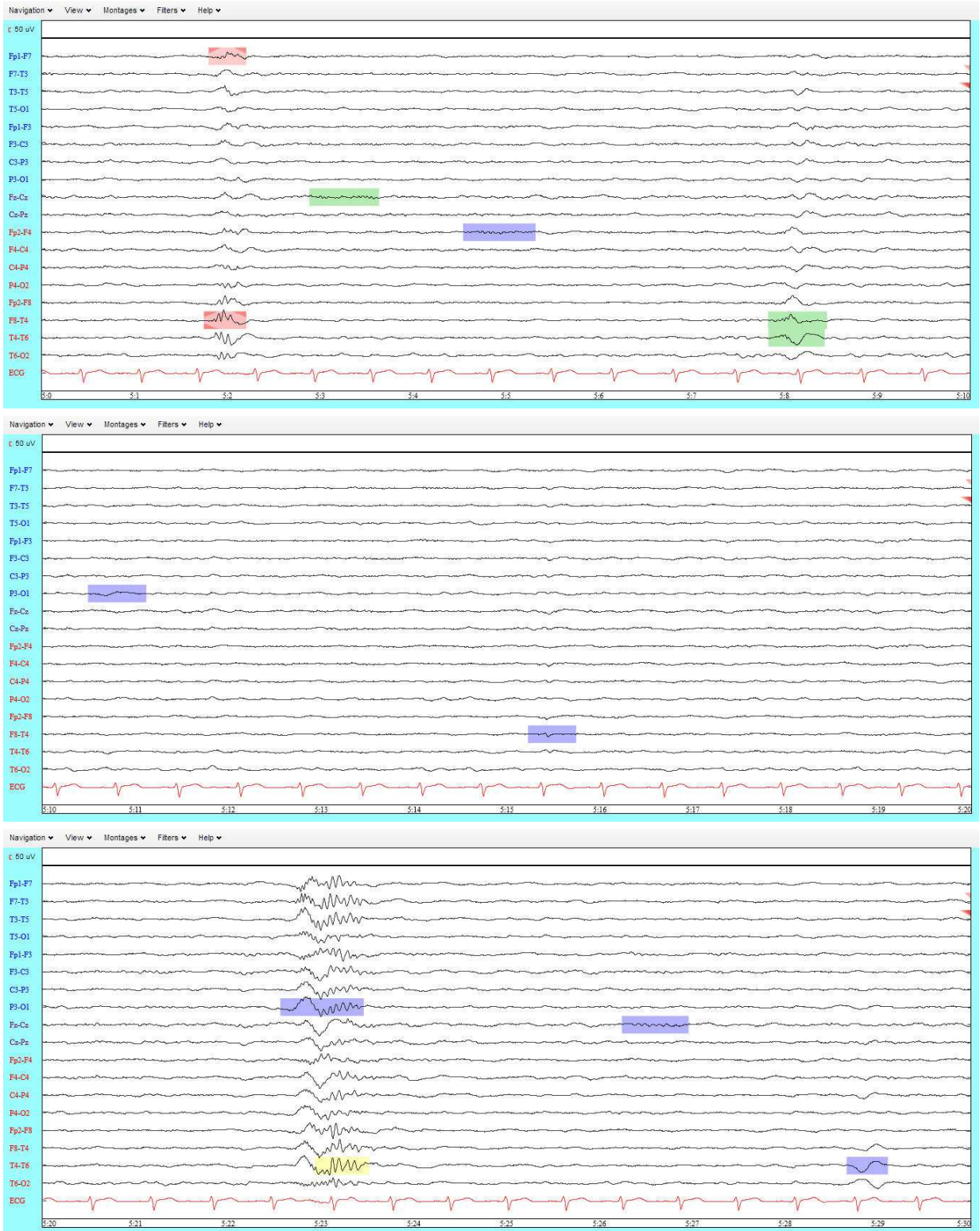


Figure 4.11: Raw yellow box candidates annotated by Simulation#2 on Patient#5

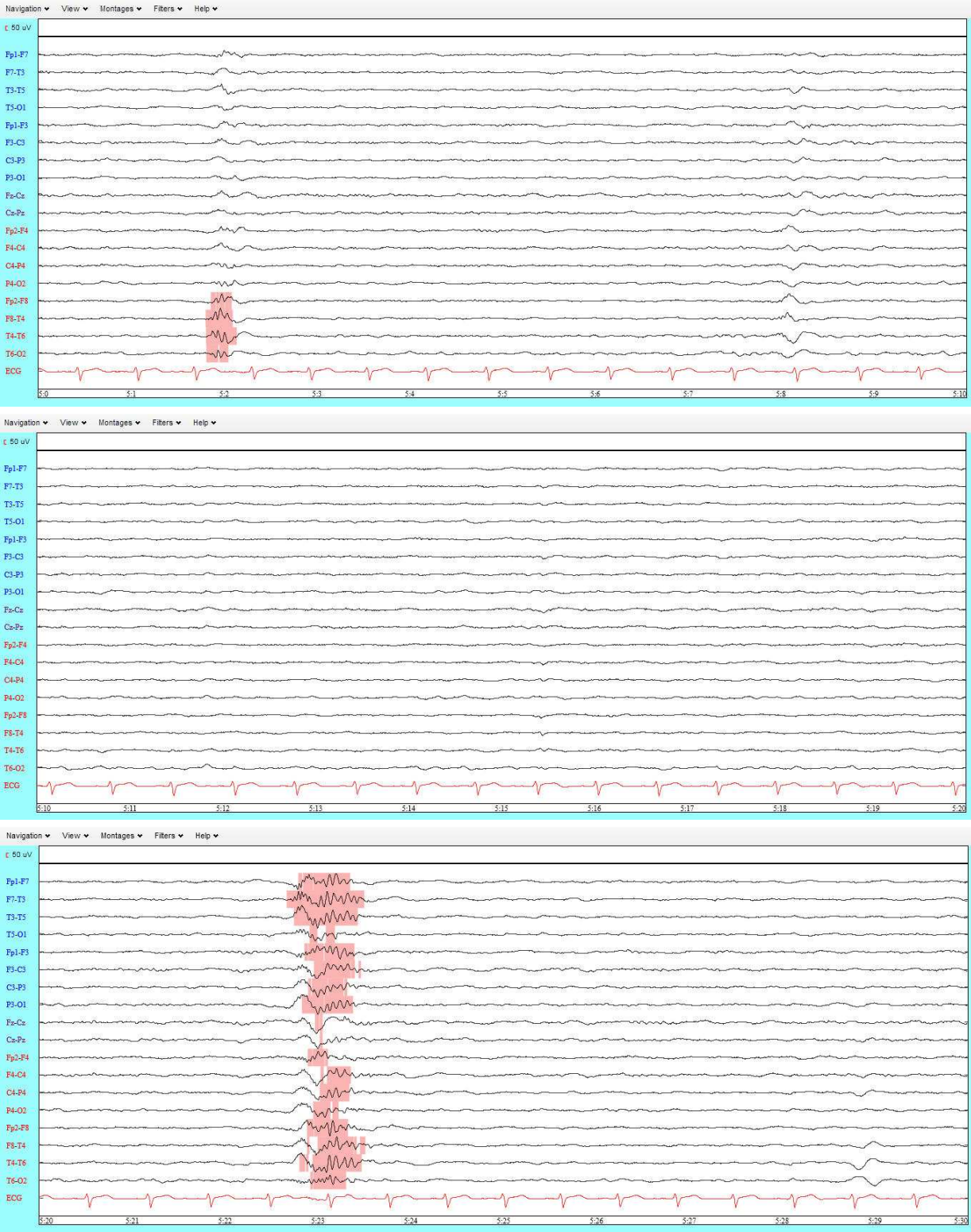


Figure 4.12: Yellow box annotated by Simulation#2 on Patient#5 after grouping

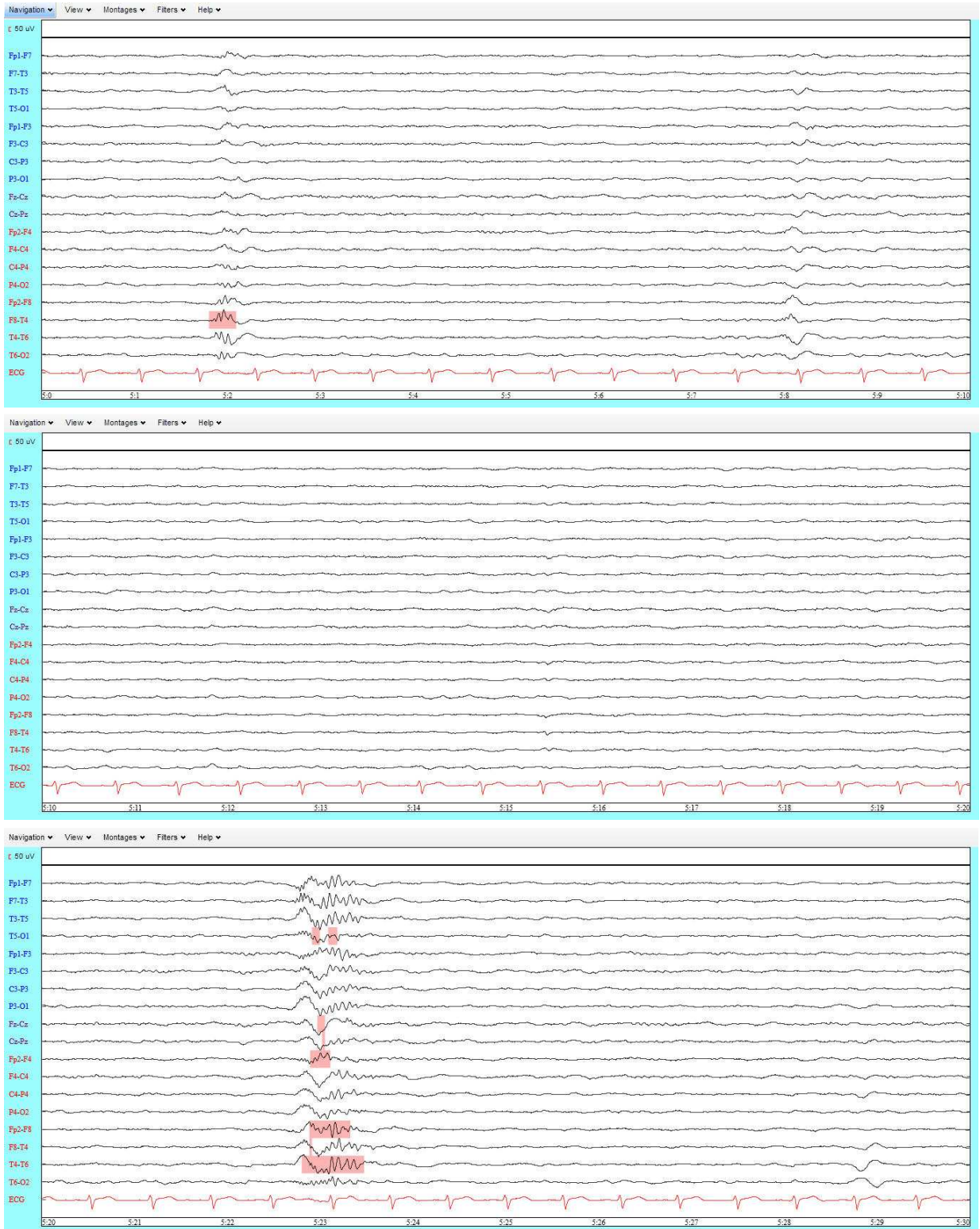


Figure 4.13: Raw yellow box candidates annotated by Simulation#9 on Patient#5

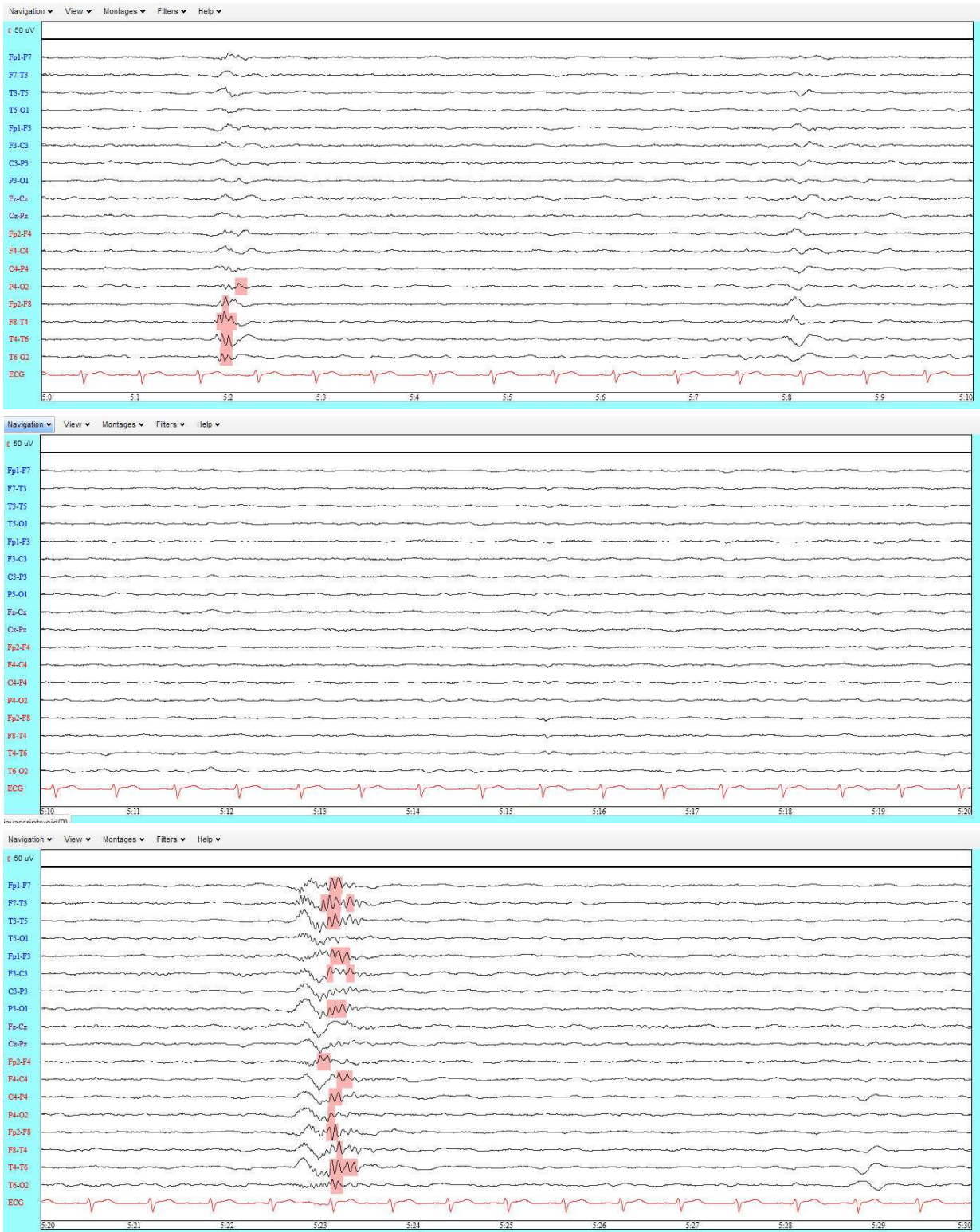
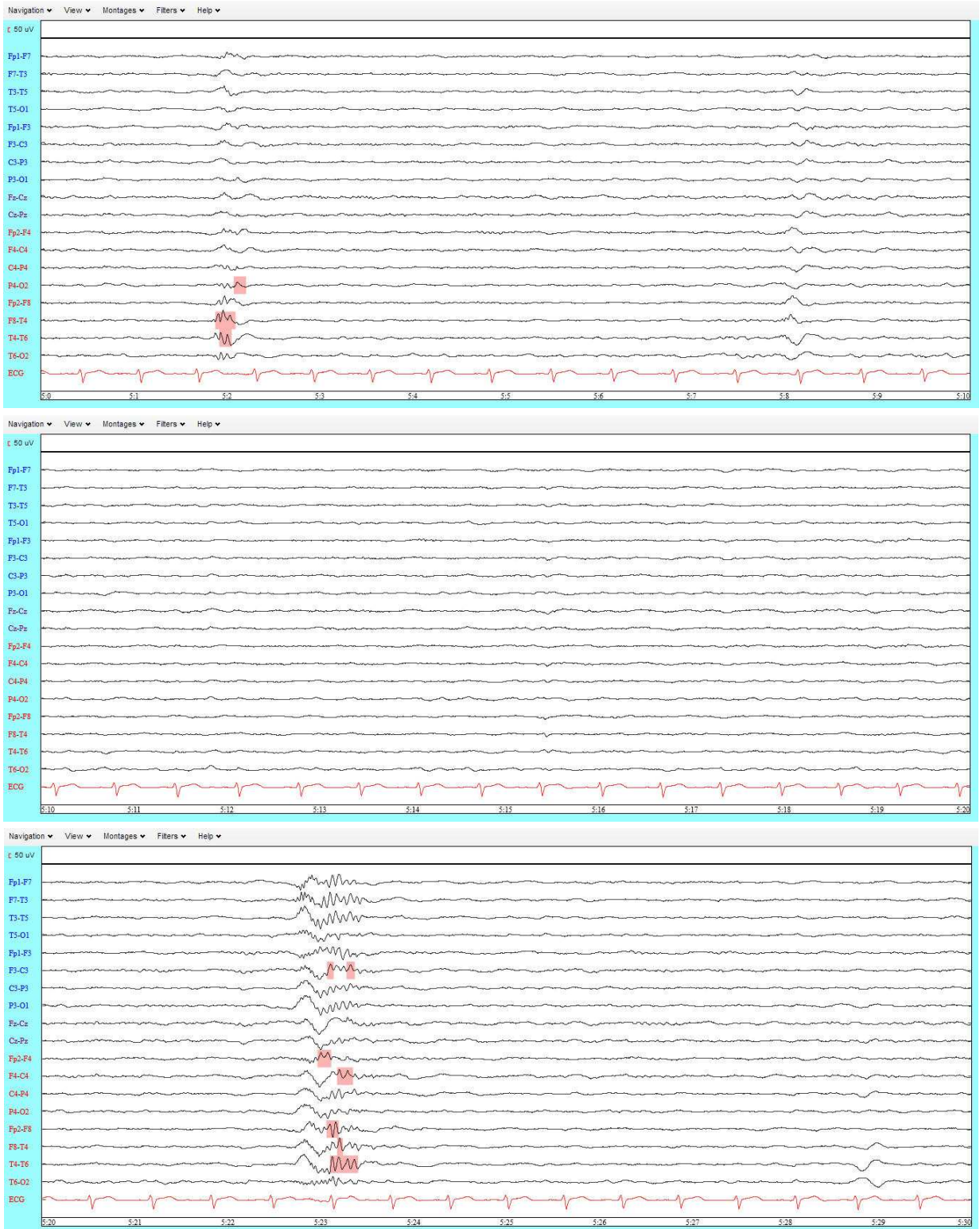


Figure 4.14: Yellow box annotated by Simulation#9 on Patient#5 after grouping



Chapter 5

Conclusions and Discussion

5.1 Yellow-Box Classification

5.1.1 Performance of crisp classification

In the crisp classification stage we designed and implemented new strategies to improve the performance of machine classifiers. From a group of 7 potential wavelet features, we derived and tested 5 distinct feature sets. We assessed classifier performance by combining features derived from two different mother wavelets. We also ran tests to determine if any improvement leading by certain strategy was statistically significant. Our results showed that our new wavelet features improve the classification ability (in the best case, +5.75% in sensitivity and +6.76% in specificity at highly significant level: $\alpha=0.01$) Our results also showed that the use of two dual-mother-wavelets in a classifier may be better than using a single-mother-wavelet under the condition that both wavelets are able to detect the events of interest. The cooperation of the new features and the dual-wavelet strategy provided a significant improvement. The classification results showed that Set#1 and Set#3 are the top 2 feature sets, while Set#1 has smaller dimension. We observed that the improvement in specificity and distance-to-(0,1) is always significant and this was confirmed by a small type II error in the power test. It is difficult to improve the sensitivity at a significant level using only one strategy. The improvement of sensitivity is higher when both strategies are used, but it is still inferior to that of specificity. Many factors contributed to this result. We think the various morphologies of ETs make it difficult to represent these signal patterns within a single feature set. The situation

might be improved if ETs are subdivided into multiple classes since there are at least 3 forms of ETs (spike, sharp wave, spike followed by slow wave). The small population of ET also had negative effect. Our previous study showed that with a large dataset, the performance of classification was improved. We think that the dual mother wavelet strategy improves performance of the machine learning classifiers because it may be difficult to fully represent these various ET signal patterns only using features from a single mother wavelet.

Many factors were observed to have positive effects on YB classification performance. Besides new wavelet features and multiple mother wavelets strategy, the inclusion of the spatial features is also contributed to a better performance: The addition of spatial features improved the classification results in all but one case. We also discussed the feasibility of reducing the dimension of the feature vectors, the necessity of keeping wavelet features in the high frequency range and the effect of increasing the cardinality of the dataset. Using only wavelet subband maxima as features degraded the classification results in some degree. Finally, our results indicate a larger set of AEP training samples improves classification performance. Since ETs have varying morphologies, a larger dataset can provide more samples of ET for machine learning. We do not know how many training datasets would be needed to provide optimal performance in this ET classification task, although we suspect it would be much larger than the dataset we used here.

5.1.2 Performance of fuzzy classification

The classification results listed in Table 3.20 indicate significant influence that the initial membership function values have on the outcomes. The fuzzy tests using membership functions that are created by taking the distribution of the confidence factors and the effects of “votes” into consideration achieved better performances than a benchmark crisp test did. It is plausible to improve the performance by fuzzy strategy with this dataset. The improvement of the sensitivity of the best fuzzy case versus the benchmark crisp case on all data (2.67%) was 2% higher than that on a dataset without the controversial data (0.67%), while both improvements of the specificity were roughly equal (2.29% and 2.34%). Notice that this is a balanced training test, while the majority of the data is nonAEP. This characteristic of the dataset determines that the AEP class is more sensitive to the quality of the training data than the nonAEP class. When the quality of the dataset is low, the fuzzy strategy provides different ranks of membership values which allows an event to analyze its neighbors with low membership values. The crisp strategy simply binarizes the data,

while a real AEP could be mistaken due to its less representative AEP neighbors. The traits of the two strategies determine that there was a large improvement of fuzzy strategy when the quality of the dataset was poor. When the quality of the dataset improves, the influence of the less representative AEP neighbors is weakened, which leads to a small improvement in sensitivity. The results of fuzzy classification also confirmed the conclusion that dual-wavelet strategy can improve the performance.

Table 3.20 also indicates when the number of nearest neighbor, ‘k’, goes up, the specificity increases and the sensitivity decreases. An explanation for this trend is that there are more nonAEP data than AEP data in the vector space and thus the classifier tends to favor the nonAEP class when more neighbors are involved in decision making. In most cases, the best performance occurs when $k = 1$ under the same condition. Three cases are chosen here for reviewing: Condition16, in which the best result occurs when $k = 1$; Condition8, in which the best result occurs when $k = 3$.

Figure 5.1 to Figure 5.5 illustrate the change of membership values in Condition16 before and after classification. The energy of the error per annotation is defined as:

$$E = \frac{1}{N} \sum^n (mbv_{pre-classification}^n - mbv_{post-classification}^n)^2 \quad (5.1)$$

where mbv^n is the membership value of the n th annotation and N is the number of annotations; N times E equals the energy of the error. When ‘k’ increases, the distribution of the membership values show a trend of “stretching”, which means the post-classification membership values are distributed more evenly in the interval $[0,1]$ than the pre-classification membership values. The stretching causes a significant reduction of the proportion of membership value greater than 0.5, which is the source of true positive and false positive. This trend indicates when ‘k’ is growing, the populations of both true positive and false positive are decreasing, which leads to a decrease in sensitivity and an increase in specificity. However, due to the fact that the cardinality of nonAEP is more than ten times larger than that of AEP, sensitivity decreases more than specificity increases, thus distance-to-(0,1) measuring the overall performance is also decreasing. Figure 5.6 demonstrates how the total energy evolves with ‘k’ in Condition16, while Figure 5.7 demonstrates the evolution of average energy per vector. Although the algorithm yields the best result at ‘k=1’, the corresponding error energy is still the highest for this ‘k’ value. The reduction of error energy at large ‘k’s is due to the raise of specificity and the large cardinality of nonAEP. Notice that in Figure 5.7 the trend of average error energy of nonAEP is almost overlapped with that of all data, which reflects the

significance of the proportion of nonAEP population.

Figure 5.8 to Figure 5.12 illustrate the change of membership values in Condition8 before and after classification. Figure 5.13 and Figure 5.14 illustrate the evolution of error energy. Condition8 shows a similar trend of evolution in the distribution of the values of membership and in the error energy. Yet Condition8 achieves the best result when $k = 3$. Unlike the smooth distribution in Condition16, Condition8 indicates the original membership values have a step-shape distribution. The membership values yielded by 1-nnr retain the step-shape trend in distribution. After implementing 1-nnr, for about one third of the nonAEP vector, the absolute differences between original membership values and updated membership values are over 0.5. Under this circumstance, the nonAEP vectors cannot become true negatives. After implementing 3 or higher nearest neighbor, the distribution of membership values is stretched; the differences between the original membership values and the updated membership values are reduced. Notice that in this condition, the overall performances of $k = 3, 5, 7$ and 9 are all better than that of $k = 1$ (listed in Table 3.20).

Notice that in Figure 5.7, the error energy per vector of Condition16 ranges from 0.065 to 0.115, where that of Condition8 in Figure 5.14 ranges from 0.09 to 0.17. This is another indication that the membership function initialization strategy used in Condition16 is superior to others’.

Figure 5.1: Exemplar of biased weights (Condition16) with $k = 1$
distribution of confidence factors between before and after implementing fuzzy 1–NNR

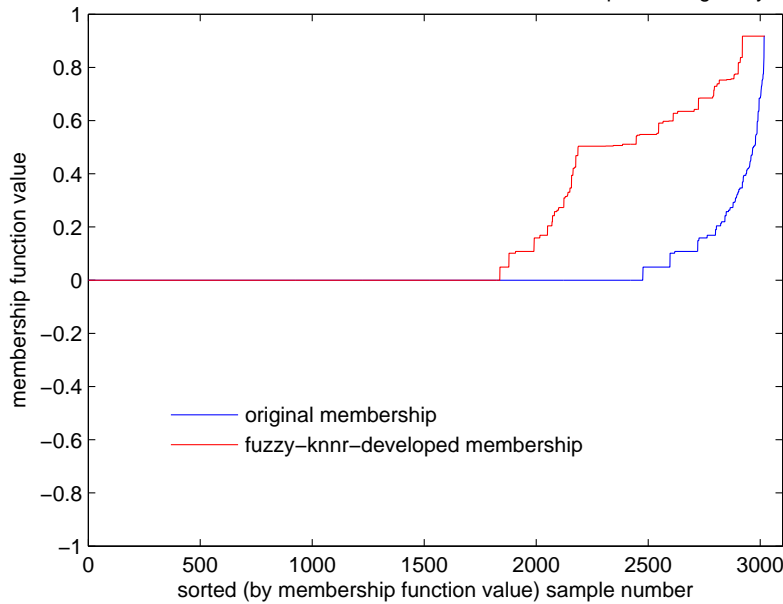


Figure 5.2: Exemplar of biased weights (Condition16) with $k = 3$

distribution of confidence factors between before and after implementing fuzzy 3–NNR

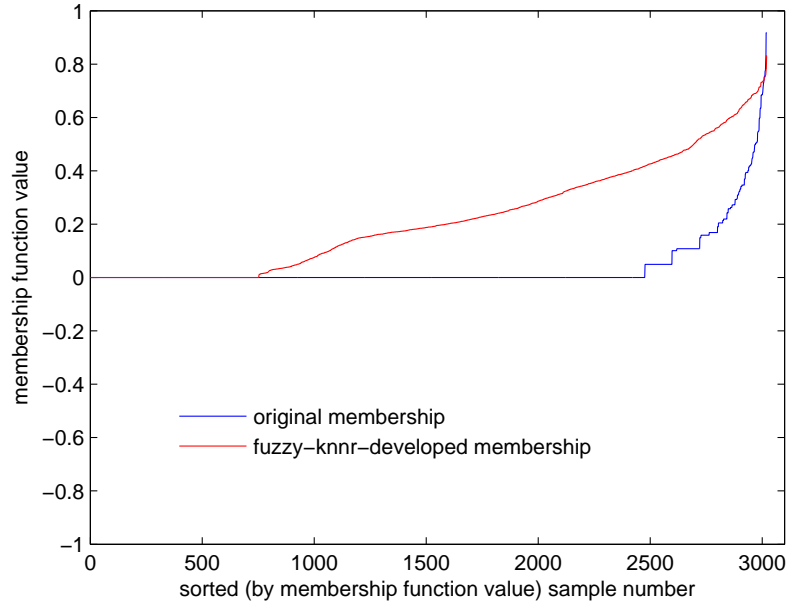


Figure 5.3: Exemplar of biased weights (Condition16) with $k = 5$

distribution of confidence factors between before and after implementing fuzzy 5–NNR

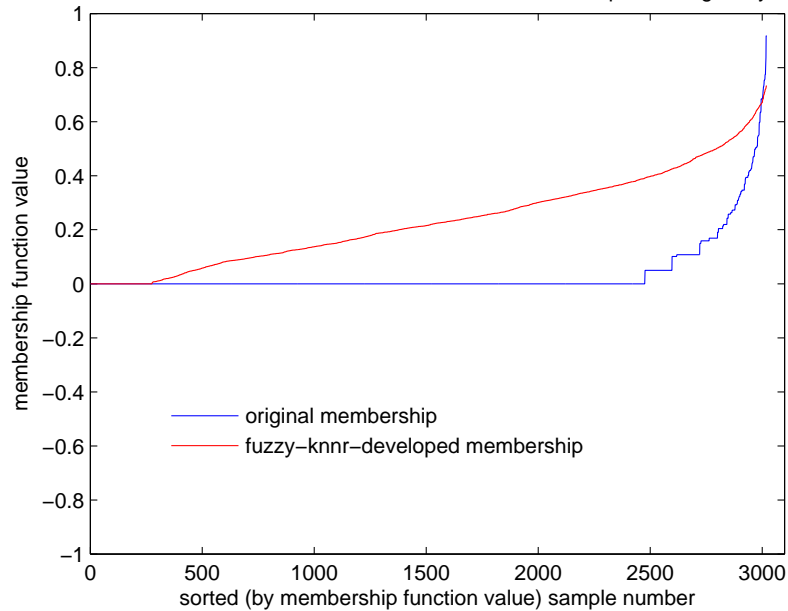


Figure 5.4: Exemplar of biased weights (Condition16) with $k = 7$
distribution of confidence factors between before and after implementing fuzzy 7-NNR



Figure 5.5: Exemplar of biased weights (Condition16) with $k = 9$
distribution of confidence factors between before and after implementing fuzzy 9-NNR

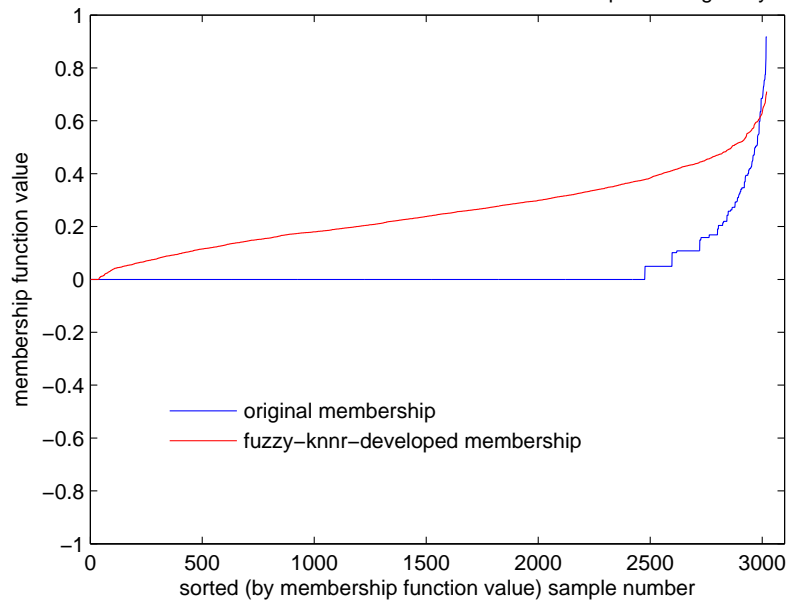


Figure 5.6: Energy of error of biased weights (Condition16) with different choice of 'k'

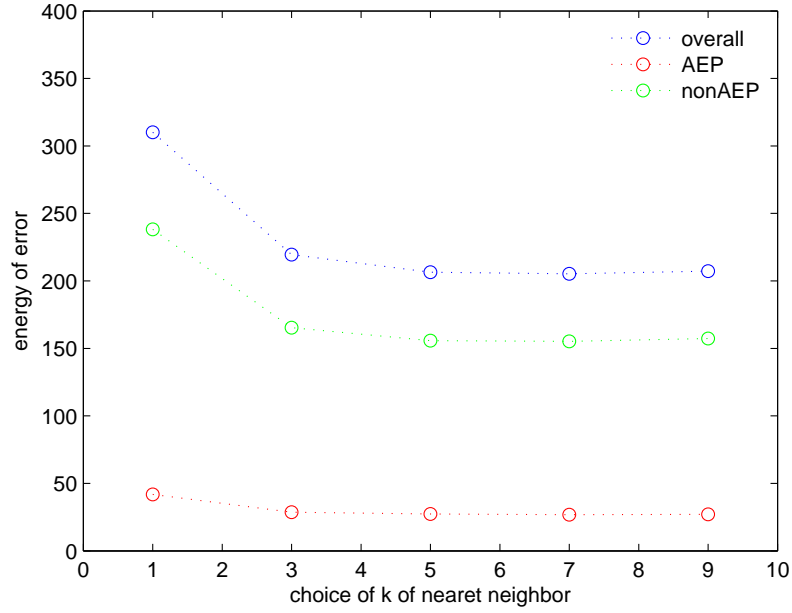


Figure 5.7: Energy of error per annotation of biased weights (Condition16) with different choice of 'k'

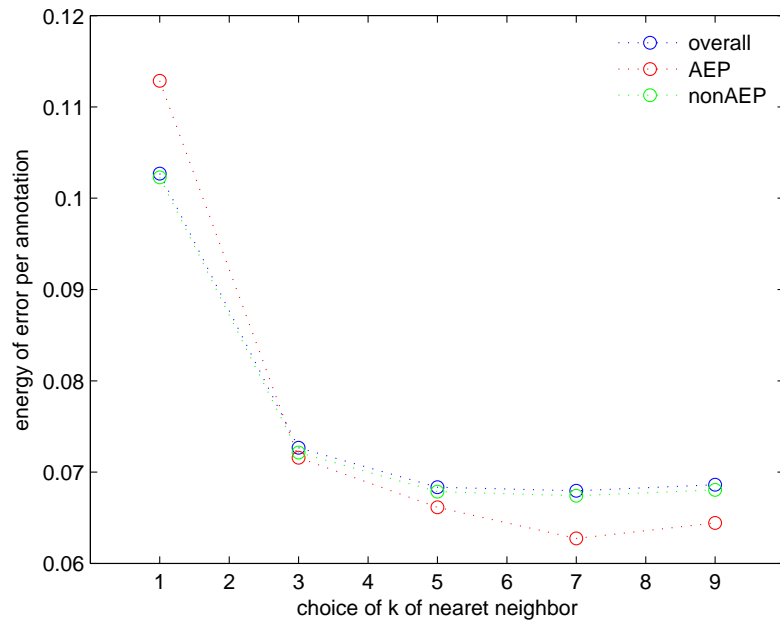


Figure 5.8: Exemplar of sigmoid transfer (Condition8) with $k = 1$

distribution of confidence factors between before and after implementing fuzzy 1–NNR

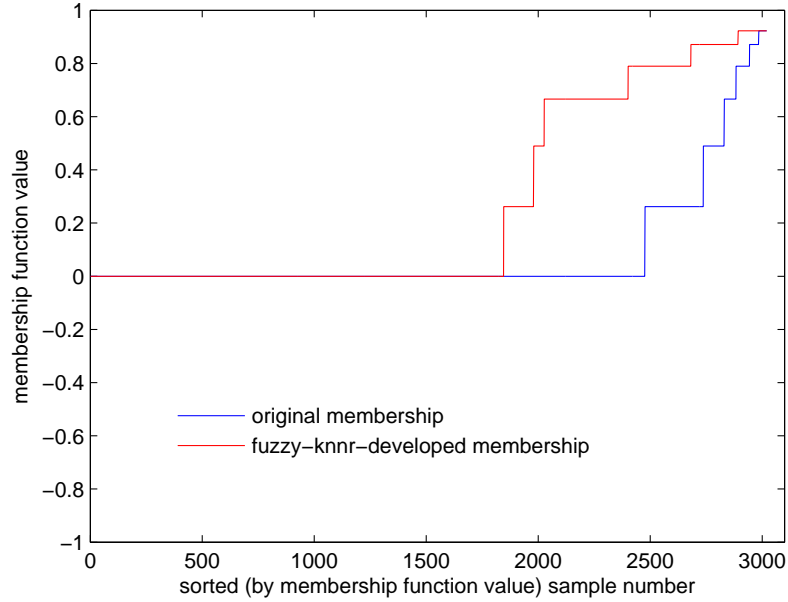


Figure 5.9: Exemplar of sigmoid transfer (Condition8) with $k = 3$

distribution of confidence factors between before and after implementing fuzzy 3–NNR

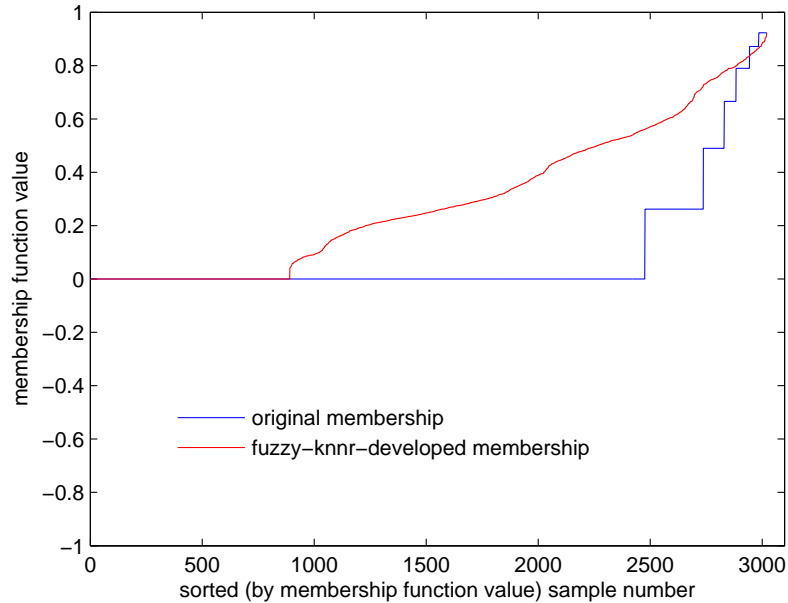


Figure 5.10: Exemplar of sigmoid transfer (Condition8) with $k = 5$

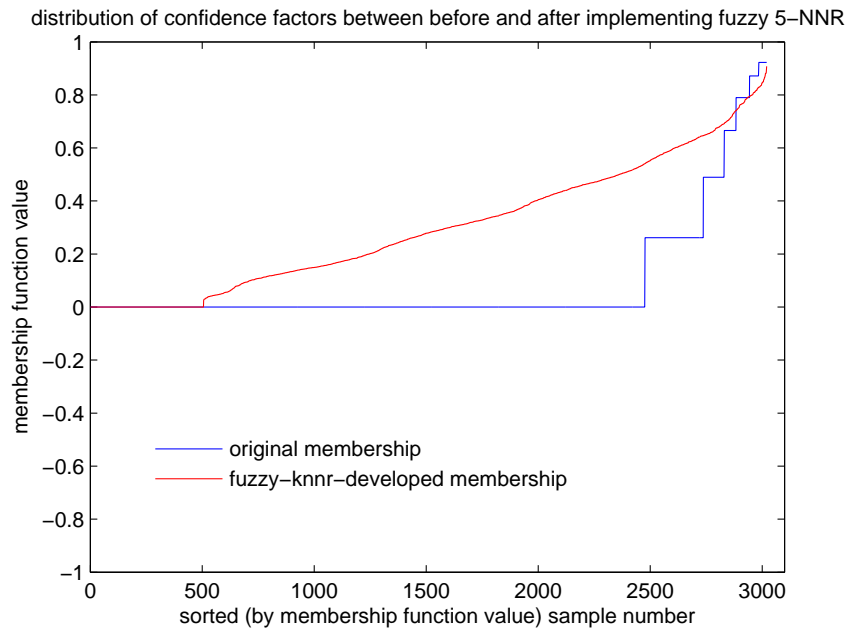


Figure 5.11: Exemplar of sigmoid transfer (Condition8) with $k = 7$

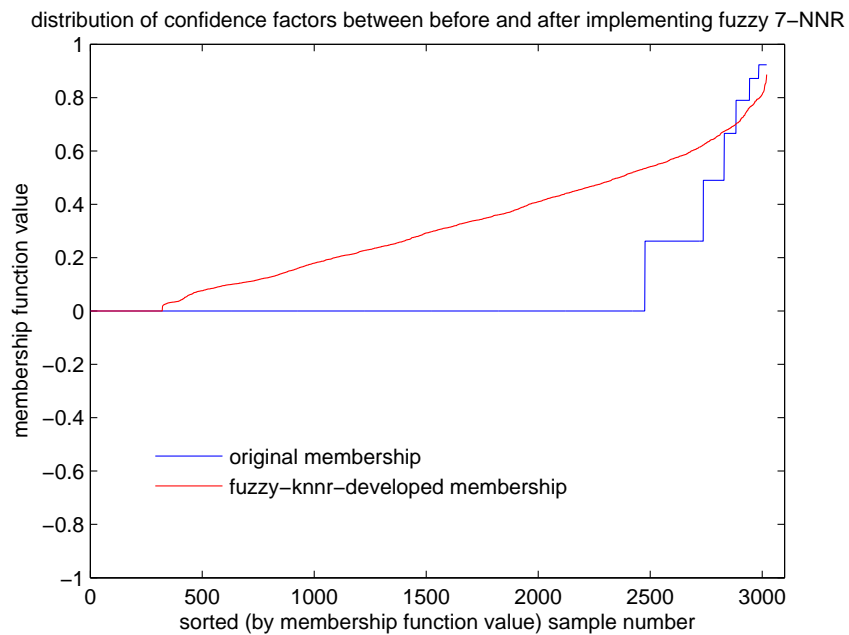


Figure 5.12: Exemplar of sigmoid transfer (Condition8) with $k = 9$



Figure 5.13: Energy of error of sigmoid transfer (Condition8) with different choice of 'k'

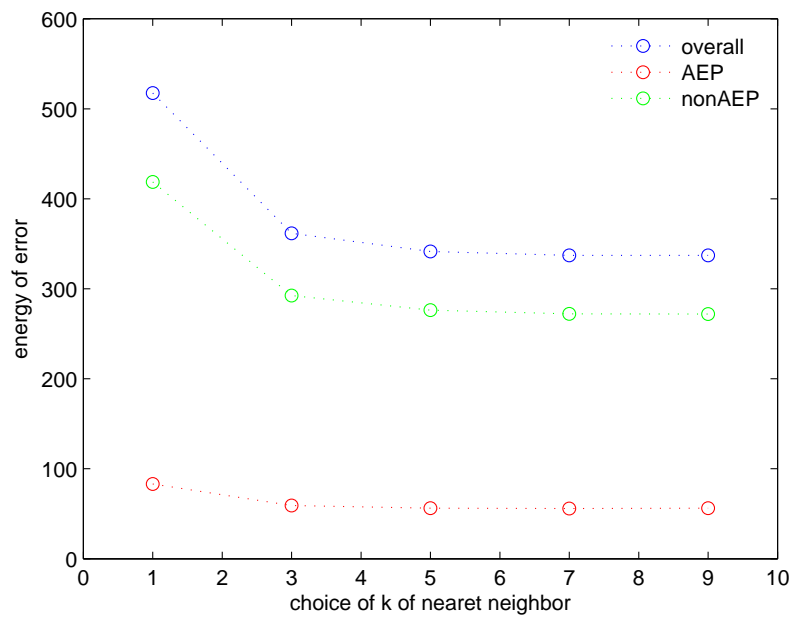
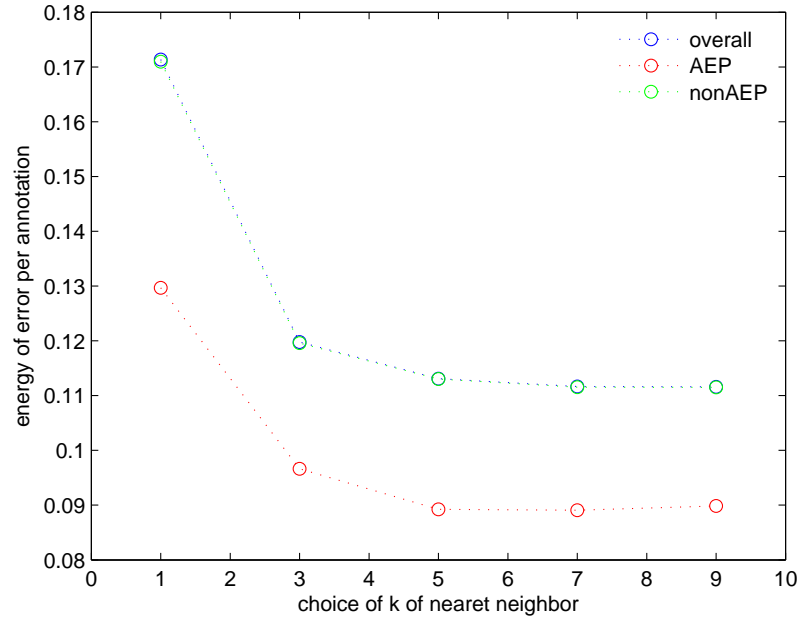


Figure 5.14: Energy of error per annotation of sigmoid transfer (Condition8) with different choice of 'k'



5.2 Yellow-Box Detection

In the YB detection part, we implemented Indiradevi's algorithm with some modification in weights. In Table 4.6 we concluded that the performance of the algorithm in detecting paroxysmal events is improved by optimization of weights. Although the specificities are inversely proportional to sensitivities, the absolute value of the decrease of specificity is not as large as the absolute value of the increment of sensitivity, which indicates the optimization of subband weights improves the overall performance. Yet optimized weights yield long length overlap, non-overlap and high cost of transformation at the same time. An explanation is that the detected annotations are relatively shorter than the expert-marked annotations and thus the former costs more to transform. The reason that the sensitivities of ETs are 30% higher than those of overall data is due to the fact that Indiradevi's algorithm favors the ETs. It is designed to detect spiky events like ETs. Some non-ETs have properties of wave rather than spike, which increase the difficulties in detection. The trade-off of higher sensitivity is a slight decrease of specificity and a severely degraded selectivity. Many factors cause the low selectivity. As mentioned in Section 2.2, the experts only place a single YB on the channel that appears to have the highest amplitude even when the paroxysmal events also

appear on other channels, which leads to many false “FP”. When the amplitudes in two channels are close, it will be difficult for human experts/machine to reach an agreement due to diversity of individual experience/algorithm.

Despite the fact that DB4 obtains the highest correlation coefficients with the epileptic spike signal, DB2 achieves better paroxysmal events detection results when using Indiradevi’s algorithm. In general, DB2 yields better sensitivity results than DB4 does while the specificities yielded by DB2 are less than 2% worse than those yielded by DB4, which suggests DB2 can be more efficient in paroxysmal events detection. We also observe that the sensitivities on bipolar AP typical montage are 20% higher than those on average referential montage using equal or optimal subband weights. The explanation is that an event having evident traits on one montage does not have to be the same on another montage, based on the fact that the paroxysmal events (contained by YB) are marked on bipolar AP typical montages and the negative events are marked on average referential montages.

Besides Indiradevi’s algorithm, we implemented neural net with two types of features. Despite the validation differences, the two types of features show certain consistency by marking similar regions as YB on EEGnet, indicating rationality of the feature choices. Besides, the neural net yields an equal error rate of 88% for the case of ET versus negative and that of 75% for the case of total paroxysmal versus negative. After pruning, the sensitivity is even higher at the cost of a poor specificity, indicating that the subband D1 contains information favoring the negative data. Notice that when the ratio of YB:negative is 2:10, the result of specificity is almost equal to that of sensitivity, indicating that the proposed features might not be representative for negative data, or background signals.

A severe problem with tests validated by focusing on data is that the similarity of data within individual subject is ignored. These tests had a relatively high evaluation result, yet tended to annotate excessively in real world EEG recordings. The full-scale real-time simulation takes this problem into consideration and adopts high proportion of negative training data. Another problem is caused by potential YBs remaining in the data. Since experts did not annotate all the events, extra penalty is added in the evaluation of selectivity. The intra-class divergence also brings challenges, especially in nonAEP class. The divergence is clearly observed in Figure 4.10, where the waveforms in different blue YBs show completely different morphology traits.

The merging process reduces the cardinality of candidates while it keeps the machine annotated regions. Table 4.12, Table 4.13 and Table 4.14 reveal the fact that merging process barely

has any influence on sensitivity and specificity yet it has a large impact on selectivity. The grouping process, on the other hand, reduces the sensitivity to a large degree, leading a slight improvement in specificity and a median improvement in selectivity. This change is caused by inconsistency between the experts' and the machine's opinions about the highest amplitude. The machine takes the entire energy in the overlap interval into consideration, while the experts focus on the amplitudes of partial data with significant traits. In particular, when a burst occurs, it will be extremely challenging to make decision by visualization.

5.3 Future Research

We have achieved significant improvement in the classification of YBs by finding suitable mother wavelet choices and feature choices. We have also shown that with appropriate initialization of membership functions, a fuzzy classification strategy can be superior to a crisp one. The significance is that the fuzzy approach provides a formalism for incorporation of the rater uncertainty. One downside is that when we adopt the gradient descent to optimize the coefficients of the confidence factors, the performance did not beat that yielded by a set of empirical coefficients. The cause is worth studying further. In the future, we will review the selections of initial values, learning rate, and training data. We will explore more plus factors to benefit the optimization, hoping to yield better results.

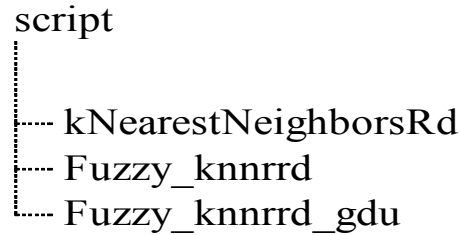
In order to increase the efficiency of the research, we have explored several strategies to detect paroxysmal events in raw rsEEGs. We have simulated real world detections. In the future, we hope to test our current detectors on new datasets and to evaluate the performance with human experts' assistance. It is also worth exploring the relationship between the wavelet-based features and the morphology-based features.

Appendices

Appendix A Matlab Code Structure

In this research, the computational work is accomplished in Matlab. Figure A.1 and Figure A.2 illustrate the most commonly used Matlab functions in this research. Table A.1 gives a brief description of each code.

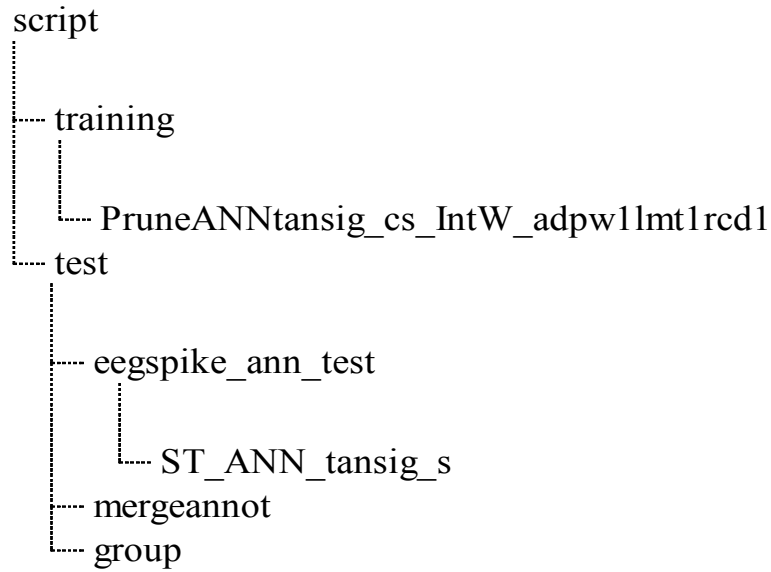
Figure A.1: Code Structure of Classification



For example:

```
.....
m=2;
desirek = [1;3;5;7;9];
.....
ds = totaltp(sort([sq1; sq2]),sn:en);
MemFunlb = MemFun(sort([sq1; sq2]),:);
.....
teIdx = CVO.test(loop);
trIdx11 = logical(trIdx2);
trial_Hid((loop-1)*subn*2*(kfd-1)+1:loop*subn*2*(kfd-1),...
          loopv+1) = trialtp(trIdx11);
trial_STid((loop-1)*(subn1+subn2)+1:loop*(subn1+subn2),...
           loopv+1) = trialtp(teIdx);
covt1 = cov(ds(trIdx11,:));
[neighbors5 distances5 testLb5] = Fuzzy_knnrrd(ds(trIdx11,:),...
        MemFunlb(trIdx11,:),ds(teIdx,:),m,desirek,covt1);
testlb(teIdx,:) = testLb5(:, :, 1);
```

Figure A.2: Code Structure of Detection



For example:

```
.....
sn=6;
en=25;
fn=en-sn+1;
Train_epoch=3000;
learnrate = 0.001;
momentum = 0;
bias = 0;
hid = 2*fn+1;
otpt =2;
Intlrt = 1e-6;
Lchg = 300;
rcdlp = 100;
rto = [1 2 10];
.....
trIdx11 = ismember( ds(:,27), record_train{1,loop} ) ;
```

```

teIdx = ismember( ds(:,27), record_test{1,loop} ) ;
Hdata = ds(trIdx11,sn:en)';
Tdata = ds(teIdx,sn:en)';
Hlabel = ones(otpt, size(Hdata,2));
Hlabel(1,record_train{2,loop}+record_train{3,loop}+1: ...
    record_train{2,loop}+record_train{3,loop}+ ...
    record_train{4,loop}) = -1;
Hlabel(2,1:record_train{2,loop}+record_train{3,loop}) = -1;
[ vdim      vnum ] = size(Hdata);
[ tdim      tnum ] = size(Hlabel);
IntW = zeros( (vdim+1)*hid+tdim*(hid+1), 1 );
for i=1:size(IntW,1)
    IntW(i,1)= rand(1)*1e-4*(-1)^randi(2,1,1) ;
end
[Nod InitWetr FinlWetr ErTgr Sij lrtrd Nodrcd Wetrcd Sijrcd] = ...
    PruneANNtansig_cs_IntW_adpw1lmt1rcd1( Hdata, Hlabel, hid, ...
    Train_epoch, 0.000000001, learnrate, bias, momentum, IntW, ...
    Intlrt, Lchg, rcdlp, Date1 );
.....

```

```

.....
DetSp = DTALLs( : , (coli-1)*25+sn : (coli-1)*25+en ) ;
[ DetSpSyn ] = eegspike_ann_test( DetSp, ...
    FinlWetr_all10(:,1,loop), hid, otpt, bias );
save( ['./rawant' num2str(Date) '/RawAnt_' wlstr ...
    'level5_Diff' num2str(ptID) '_' num2str(win_wt) ...
    'w_BPAPT' num2str(bpid) '.txt'], 'DetSpSyn', '-ASCII')
DetSpSyn = mergeannot( DetSpSyn, annotMR, annotDc );
.....

```



```

function [ OLabel ] = eegspike_ann_test( IPn, FinlW, hdn, tdim, bias )
    OLabel = zeros( size(IPn,1), size(FinlW,2) );
    for j = 1:size(FinlW,2)
        [ Label ] = ST_ANN_tansig_s( IPn', FinlW(:,j), hdn, tdim, bias );
        for i = 1:size(IPn,1)
            [ val sqc ] = max(Label(i,:));
            if sqc~=tdim && val>0
                OLabel(i,j)=1;
            end
        end
    end
end
j
end

```

Table A.1: Selected Matlab Code

FILE NAME	CONTENT	TYPE
kNearestNeighborsRd.m	crisp k-NNR algorithm using Mahalanobis distance	function
Fuzzy_knnrrd.m	fuzzy k-NNR algorithm using Mahalanobis distance	function
Fuzzy_knnrrd_gdu.m	optimization of coefficients of confidence factor using gradient descent strategy	function
PruneANNtansig_cs_IntW_adpw1lmt1rcd1.m	training of ANN, including computation of S_{ij} and adaptation of learning rate	function
eegspike_ann_test.m	YB detection on multiple channels	function
ST_ANN_tansig_s.m	YB detection using trained ANN	function
mergeannot.m	merging and discarding of YBs	function

Bibliography

- [1] N. Acir, I. Oztura, M. Kuntalp, B. Baklan, and C. Guzelis. Automatic detection of epileptiform events in eeg by a three-stage procedure based on artificial neural networks. *IEEE Transactions on Biomedical Engineering*, 52:30–40, January 2005.
- [2] H. Adeli, Z. Zhou, and N. Dadmehr. Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123:69–87, February 2003.
- [3] M. Adjouadi, M. Cabrerizo, M. Ayala, D. Sanchez, I. Yaylali, P. Jayakar, and A. Barreto. A new mathematical approach based on orthogonal operators for the detection of interictal spikes in epileptogenic data. *Biomedical sciences instrumentation*, 40:175–180, 2004.
- [4] J. S. Barlow. EEG transient detection by matched inverse digital filtering. *Electroencephalography and Clinical Neurophysiology*, 48:246–248, February 1980.
- [5] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks*, 5:537–550, July 1994.
- [6] L. Behera, S. Kumar, and A. Patnaik. On adaptive learning rate that guarantees convergence in feedforward networks. *IEEE Transactions on Neural Networks*, 17:1116–1125, 2006.
- [7] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1998.
- [8] R. L. Cannon, J. V. Dave, and J. C. Bezdek. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:248–255, 1986.
- [9] L. Chen and H. L. Tang. Improved computation of beliefs based on confusion matrix for combining multiple classifiers. *Electronics Letters*, 40:238–239, February 2004.
- [10] R. Cooper, J. W. Osselton, and J. C. Shaw. *EEG Technology*. Butterworths, 1969.
- [11] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, November 1988.
- [12] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
- [13] B. L. Davey, W. R. Fright, G. J. Carroll, and R. D. Jones. Expert system approach to detection of epileptiform activity in the EEG. *Medical & Biological Engineering & Computing*, 27:365–370, July 1989.
- [14] P. Guedes de Oliveira, C. Queiroz, and F. Lopes da Silva. Spike detection based on a pattern recognition approach using a microcomputer. *Electroencephalogr Clin Neurophysiol*, 56, July 1983.

- [15] A. Dingle, R. Jones, G. Carroll, and W. Fright. A multistage system to detect epileptiform activity in the eeg. *IEEE Transactions on Biomedical Engineering*, 40:1260–1268, December 1993.
- [16] C. Faure. Attributed strings for recognition of epileptic transients in EEG. *International Journal of Bio-Medical Computing*, 16, May 1985.
- [17] M. Feucht, K. Hoffmann, K. Steinberger, H. Witte, F. Benninger, M. Arnold, and A. Doering. Simultaneous spike detection and topographic classification in pediatric surface EEGs. *Neuroreport*, 8:2193–2197, July 1997.
- [18] G. Fischer, N. J. I. Mars, and F. H. Lopes da Silva. Pattern recognition of epileptiform transients in the electroencephalogram. *Prog. Rep. Institute of Medical Physics, Utrecht, The Netherlands*, 7:22–31, 1980.
- [19] D. Flanagan, R. Agarwal, Y. H. Wang, and J. Gotman. Improvement in the performance of automated spike detection using dipole source features for artefact rejection. *Clinical Neurophysiology*, 114:38–49, January 2003.
- [20] N. B. Fountain and J. M. Freeman. EEG is an essential clinical tool: Pro and con. *Epilepsia*, 47:23–25, October 2006.
- [21] J. D. Frost. Automatic recognition and characterization of epileptiform discharges in the human EEG. *Clinical Neurophysiology*, 2:231–249, July 1985.
- [22] J. R. Glover, N. Raghavan, P. Y. Ktonas, and J. D. Frost. Context-based automated detection of epileptogenic sharp transients in the EEG: elimination of false positives. *IEEE Transactions on Biomedical Engineering*, 36:519–527, June 1989.
- [23] H. Goelz, R. D. Jones, and P. J. Bones. Wavelet analysis of transient biomedical signals and its application to detection of epileptiform activity in the EEG. *Clin Electroencephalogr*, 31:181–191, October 2000.
- [24] J. Gotman and P. Gloor. Automatic recognition and quantification of interictal epileptic activity in the human scalp EEG. *Electroencephalogr Clin Neurophysiol*, 41, November 1976.
- [25] Inan Guler and Elif Derya Ubeyli. Adaptive neuro-fuzzy inference system for classification of eeg signals using wavelet coefficients. *Journal of Neuroscience Methods*, 148:113–121, April 2005.
- [26] J. J. Halford. Computerized epileptiform transient detection in the scalp electroencephalogram: Obstacles to progress and the example of computerized eeg interpretation. *Clinical Neurophysiology*, 120:1909–1915, October 2009.
- [27] J. J. Halford, R. J. Schalkoff, J. Zhou, S. R. Benbadis, W. O. Tatum, R. P. Turner, S. R. Sinha, N. B. Fountain, A. Arain, P. B. Pritchard, E. Kutluay, G. Martz, J. C. Edwards, C. Waters, and B. C. Dean. Standardized database development for EEG epileptiform transient detection: EEGnet scoring system and machine learning analysis. *Journal of Neuroscience Methods*, 212:308–316, January 2013.
- [28] W. A. Hauser and D. C. Hesdorffer. *Epilepsy: frequency, causes and consequences*. Demos, 1990.
- [29] Matlab help file.
- [30] W. E. Hostetler, H. J. Doller, and R. W. Homan. Assessment of a computer program to detect epileptiform spikes. *Electroencephalogr Clin Neurophysiol*, 83:1–11, July 1992.

- [31] K. P. Indiradevi, E. Elias, and P. S. Sathidevi. Automatic detection of epileptic spikes in the long term electroencephalogram using wavelet transform. *Conference on Computational Intelligence and Multimedia Applications*, 1:552–556, December 2007.
- [32] K. P. Indiradevi, E. Elias, P. S. Sathidevi, S. Dinesh Nayak, and K. Radhakrishnan. A multi-level wavelet approach for automatic detection of epileptic spikes in the electroencephalogram. *Computers in Biology and Medicine*, 38:805–816, April 2008.
- [33] D. Kahaner, C. B. Moler, and S. Nash. *Numerical methods and software*. Prentice Hall, 1989.
- [34] T. Kalayci and O. Ozdamar. Wavelet preprocessing for automated neural network detection of EEG spikes. *IEEE Engineering in Medicine and Biology Magazine*, 14:160–166, March 1995.
- [35] E. D. Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks*, 1:239–242, 1990.
- [36] J. M. Keller, M. R. Gray, and J. A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 15:580–585, 1985.
- [37] L.G. Kiloh. *Clinical Electroencephalography*. Butterworths, 1981.
- [38] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On combining classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, March 1998.
- [39] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence(IJCAI)*, 1995.
- [40] P. Y. Ktonas. Automated spike and sharp wave (ssw) detection. *Methods of analysis of brain electrical and Magnetic signals. EEG handbook*, 1:211–241, 1987.
- [41] H. Liu, T. Zhang, and F. Yang. A multistage multimethod approach for automatic detection and classification of epileptiform eeg. *IEEE Transactions on Biomedical Engineering*, 49:1557–1566, December 2002.
- [42] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Transactions of the American Mathematical Society*, 315:69–87, September 1989.
- [43] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, July 1989.
- [44] M. C. Mozer and P. Smolensky. Skeletonization: a technique for trimming the fat from a network via relevance assessment. *Advances in neural information processing systems*, pages 107–115, 1989.
- [45] E. Niedermeyer and F. Lopes Da. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Williams & Wilkins, 1993.
- [46] I. Omerhodzic, S. Avdakovic, A. Nuhanovic, K. Dizdarevic, and K. Rotim. Energy distribution of EEG signal components by wavelet transform. *Wavelet Transforms and Their Recent Applications in Biology and Geoscience*, pages 45–60, 2012.
- [47] O. Ozdamar and T. Kalayci. Detection of spikes with artificial neural networks using raw eeg. *Computers and Biomedical Research*, 31:122–142, 1998.
- [48] H. S. Park, Y. H. Lee, N. G. Kim, D. S. Lee, and S. I. Kim. Detection of epileptiform activities in the EEG using neural network and expert system. *Studies in health technology and informatics*, 52:1255–1259, 1998.

- [49] H. S. Park, Y. H. Lee, D. S. Lee, and S. I. Kim. Detection of epileptiform activity using wavelet and neural network. In *Engineering in Medicine and Biology Society, Proceedings of the 19th Annual International Conference of the IEEE*, pages 1194–1197, 1997.
- [50] R. Polikar. The engineer’s ultimate guide to wavelet analysis: The wavelet tutorial. <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>.
- [51] S. Sanei and J. A. Chambers. *EEG Signal Processing*. John Wiley & Sons Ltd, 2007.
- [52] R. Sankar and J. Natour. Automatic computer analysis of transients in eeg. *Computers in biology and medicine*, 22:407–422, 1992.
- [53] Robert J. Schalkoff. *Pattern Recognition, statistical, structural and neural approaches*. John Wiley & Sons, Inc, 1992.
- [54] Robert J. Schalkoff. *Artificial Neural Networks*. The McGraw-Hill Companies, Inc, 1997.
- [55] L. Senhadji, J. Dillenseger, F. Wendling, C. Rocha, and A. Kinie. Wavelet analysis of EEG for three-dimensional mapping of epileptic events. *Ann Biomed Eng*, 23:543–552, 1995.
- [56] American Electroencephalographic Society. Guideline seven: a proposal for standard montages to be used in clinical EEG. *J Clin Neurophysiol*, 11:30–36, January 1994.
- [57] J. R. Stevens, B. L. Lonsbury, and S. L. Goel. Seizure occurrence and interspike interval telemetered electroencephalogram studies. *Arch Neurol*, 26:409–419, May 1972.
- [58] T. Pietila T, S. Vapaakoski, U. Nousiainen, A. Varri, H. Frey, V. Hakkinen, and Y. Neuvo. Evaluation of a computerized system for recognition of epileptic activity during long-term EEG recording. *Electroencephalography and Clinical Neurophysiology*, 90:438–443, June 1994.
- [59] V. Tresp and M. Taniguchi. Combining estimators using non-constant weighting functions. *Neural Information Processing Systems*, pages 419–426, 1994.
- [60] C. A. van Donselaar, R. J. Schimsheimer, A. T. Geerts, and A. C. Declerck. Value of the electroencephalogram in adult patients with untreated idiopathic first seizures. *Arch Neurol*, 49:231–237, March 1992.
- [61] W. R. S. Webber, B. Litt, K. Wilson, and R. P. Lesser. Practical detection of epileptiform discharges (eds) in the eeg using an artificial neural network: a comparison of raw and parameterized eeg data. *Electroencephalography and Clinical Neurophysiology*, 91:194–204, 1994.
- [62] S. B. Wilson and R. Emerson. Spike detection: a review and comparison of algorithms. *Clinical Neurophysiology*, 113:1873–1881, December 2002.
- [63] S. B. Wilson, C. A. Turner, R. G. Emerson, and M. L. Scheuer. Spike detection ii: automatic, perception-based detection and clustering. *Clinical Neurophysiology*, 110:404–411, March 1999.
- [64] M. P. Windham. Cluster validity for the fuzzy c-means clustering algorithm. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 4:357–363, 1982.
- [65] H. Witte, L.D. Iasemidis, and B. Litt. Special issue on epileptic seizure prediction. *IEEE Transactions on Biomedical Engineering*, 50:537–539, May 2003.
- [66] G. Xu, J. Wang, Q. Zhang, and J. Zhu. An automatic EEG spike detection algorithm using morphological filter. In *IEEE International Conference on Automation Science and Engineering, 2006. CASE '06.*, pages 170–175, October 2006.
- [67] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

- [68] J. Zhou, R. J. Schalkoff, B. C. Dean, and J. J. Halford. Morphology-based wavelet features and multiple mother wavelet strategy for spike classification in eeg signals. In *Engineering in Medicine and Biology Society (EMBS), 2012 Annual International Conference of the IEEE*, pages 3959–3962, 2012.